



## TESIS DOCTORAL

# Métodos estadísticos y computacionales en el estudio de tiempos de respuesta de reconocimiento de palabras

*Programa de Doctorado en Lectura y  
Comprensión*

Valencia, Marzo de 2021

Javier Breithaupt Marqués

Dirección:

Dr. Manuel Perea Lara

Dr. Pablo Gómez

Dra. Ana Marcet Herranz





Los Doctores Manuel Perea Lara (Universitat de València), Pablo Gómez (California State University San Bernardino, Palm Desert) y Ana Marcet Herranz (Universitat de València)

DECLARAMOS:

Que el trabajo titulado "Métodos estadísticos y computacionales en el estudio de tiempos de respuesta de reconocimiento de palabras" que presenta Javier Breithaupt Marqués para la obtención del título de doctor, se ha realizado bajo nuestra dirección y tutoría, cumpliendo todos los requisitos para poder optar a su lectura como Tesis Doctoral en la Universitat de València.

Y para que así conste y tenga los efectos oportunos, firmamos el presente documento.

Manuel Perea Lara      Pablo Gómez      Ana Marcet Herranz

# Índice general

<b>I</b>	<b>Presentación</b>	<b>1</b>
<b>1</b>	<b>Introducción</b>	<b>3</b>
<b>II</b>	<b>Introducción a los métodos computacionales de tratamiento de datos en tiempos de respuesta de reconocimiento de palabras</b>	<b>9</b>
<b>2</b>	<b>Curvas de supervivencia promedio y punto de divergencia mediante bootstrap e análisis de tiempos de respuesta de reconocimiento de palabras</b>	<b>11</b>
2.1	Introducción . . . . .	11
2.2	Resumen del método teórico . . . . .	12
2.3	Simulaciones computacionales . . . . .	18
2.3.1	Simulaciones realistas con poca asimetría positiva . . . . .	20
2.3.1.1	Efecto sobre la media . . . . .	20
2.3.2	Simulaciones realistas con mucha asimetría positiva . . . . .	20
2.3.2.1	Sin efecto . . . . .	22
2.3.2.2	Efecto sobre la media . . . . .	22
2.3.2.3	Estudio de la mediana con efecto sobre la media . . . . .	22
2.3.2.4	Efecto sobre la cola . . . . .	25
2.3.2.5	Estudio de la mediana con efecto sobre la cola . . . . .	25
2.4	Conclusiones . . . . .	25
<b>3</b>	<b>Are divergence point analyses suitable for response time data?</b>	<b>29</b>
3.1	Abstract . . . . .	29
3.2	Are divergence point analyses suitable for response time data? . . . . .	30

3.3	Construct validity of divergence points . . . . .	32
3.3.1	Trivial divergence points . . . . .	32
3.3.2	Valid divergence points . . . . .	36
3.4	Statistical properties of divergence point analyses . . . . .	37
3.4.1	Spurious divergence points . . . . .	38
3.4.2	Divergence points that are CDF crossings . . . . .	40
3.5	Conclusions . . . . .	44
3.6	Open practices statement . . . . .	46
3.7	References . . . . .	46
<b>4</b>	<b>Redes neuronales aplicadas al reconocimiento de palabras. Un modelo autoorganizado de aprendizaje lector</b>	<b>51</b>
4.1	Fundamentación del modelo . . . . .	51
4.2	Datos . . . . .	52
4.3	Algoritmos y código fuente . . . . .	57
4.3.1	Algoritmo original . . . . .	57
4.3.2	Códigos fuente . . . . .	59
4.4	Experimento 1. Generación aleatoria de los pesos de los estímulos.	75
4.4.1	Mapa autoorganizado: letras . . . . .	75
4.4.1.1	Comparación de parejas minúscula / mayúscula	75
4.4.1.2	Mapa de letras entrenado. Resultado global. . .	77
4.4.2	Mapa autoorganizado: palabras . . . . .	79
4.4.2.1	Comparación de parejas minúscula / mayúscula	79
4.4.2.2	Mapa de palabras entrenado. Resultado global. .	79
4.5	Experimento 2. Generación de los pesos los estímulos por representación visual. . . . .	79
4.5.1	Mapa autoorganizado: letras . . . . .	79
4.5.1.1	Comparación de parejas minúscula / mayúscula	79
4.5.1.2	Mapa de letras entrenado. Resultado global. . .	83
4.5.2	Mapa autoorganizado: palabras . . . . .	83
4.5.2.1	Comparación de parejas minúscula / mayúscula	83
4.5.2.2	Mapa de palabras entrenado. Resultado global. .	86
4.6	Método 3. Generación de los pesos de los pesos de los estímulos por representación visual. Incorporación del elemento fonético. .	86
4.6.1	Mapa autoorganizado: letras . . . . .	86
4.6.1.1	Comparación de parejas minúscula / mayúscula	86
4.6.1.2	Mapa de letras entrenado. Resultado global. . .	88

4.6.2	Mapa autoorganizado: palabras . . . . .	88
4.6.2.1	Comparación de parejas minúscula / mayúscula . . . . .	88
4.6.2.2	Mapa de palabras entrenado. Resultado global. . . . .	88
4.7	Error . . . . .	88
4.7.1	Letras . . . . .	91
4.7.2	Palabras . . . . .	93
4.8	Conclusiones . . . . .	95

### **III Tratamiento de outliers en Tiempos de Respuesta de reconocimiento de palabras 97**

#### **5 Generación datos en tiempos de respuesta de reconocimiento de palabras con presencia de outliers y posibilidades de transformación de los mismos 99**

5.1	Introducción . . . . .	99
5.2	Planteamiento del problema y Modelo Lineal Generalizado Mixto . . . . .	103
5.3	Generación de los datos . . . . .	105
5.4	Distribución de datos . . . . .	105
5.4.1	Efectos sobre la media . . . . .	106
5.4.2	Efectos sobre la cola . . . . .	108
5.5	Análisis de dispersión de los datos . . . . .	110
5.5.1	Efectos sobre la media . . . . .	112
5.5.2	Efectos sobre la cola . . . . .	112
5.6	Conclusiones . . . . .	113

#### **6 Tratamiento de Outliers en Tiempos de Respuesta de reconocimiento de palabras mediante transformaciones de datos y análisis de resultados mediante Modelos Lineales 115**

6.1	Introducción . . . . .	115
6.1.1	Resumen conceptual del Modelo Lineal . . . . .	116
6.1.2	Planteamiento del problema y Modelo Lineal . . . . .	117
6.2	Análisis de potencia . . . . .	119
6.2.1	Análisis error tipo I . . . . .	120
6.2.2	Efectos sobre la media . . . . .	120
6.2.3	Efectos sobre la cola . . . . .	122
6.3	Análisis de residuos . . . . .	124

6.3.1	Sin efecto sobre ningún parámetro ni variabilidad entre sujetos . . . . .	132
6.3.2	Efecto sobre la media y sin variabilidad entre sujetos . . .	132
6.3.3	Efecto sobre la cola y sin variabilidad entre sujetos . . . .	135
6.4	Conclusiones. . . . .	140
<b>7</b>	<b>Tratamiento de Outliers en Tiempos de Respuesta de reconocimiento de palabras mediante transformaciones de datos y análisis de resultados mediante Modelos Lineales Generalizados Mixtos</b>	<b>149</b>
7.1	Introducción y objetivos . . . . .	149
7.2	Modelo Lineal frente a Modelo Lineal Generalizado Mixto . . . .	150
7.2.1	Modelo Lineal . . . . .	151
7.2.2	Modelo Lineal Generalizado Mixto . . . . .	152
7.3	Planteamiento del problema y Modelo Lineal Generalizado Mixto	153
7.4	Análisis de potencia . . . . .	156
7.4.1	Análisis error tipo I . . . . .	160
7.4.2	Efectos sobre la media . . . . .	160
7.4.3	Efectos sobre la cola . . . . .	161
7.5	Análisis de residuos . . . . .	163
7.5.1	Sin efecto sobre ningún parámetro ni variabilidad entre sujetos . . . . .	173
7.5.2	Efecto sobre la media y sin variabilidad entre sujetos . . .	173
7.5.3	Efecto sobre la cola y sin variabilidad entre sujetos . . . .	177
7.6	Conclusiones y posibles futuras vías de trabajo. . . . .	179
7.6.1	Análisis de potencia . . . . .	179
7.6.2	Análisis de resíduos . . . . .	182
<b>IV</b>	<b>Conclusiones</b>	<b>189</b>
<b>8</b>	<b>Resumen métodos y conclusiones</b>	<b>191</b>
8.1	Presentación . . . . .	191
8.2	Bloque I . . . . .	191
8.2.1	Survival . . . . .	192
8.2.1.1	Objetivo . . . . .	192
8.2.1.2	Inicialización . . . . .	192
8.2.1.3	Método . . . . .	192

8.2.1.4	Resultados . . . . .	192
8.2.2	Redes neuronales . . . . .	193
8.2.2.1	Objetivo . . . . .	193
8.2.2.2	Inicialización . . . . .	193
8.2.2.3	Método . . . . .	194
8.2.2.3.1	Algoritmo base . . . . .	194
8.2.2.3.2	Experimentos . . . . .	194
8.2.2.4	Resultados . . . . .	195
8.3	Bloque II . . . . .	196
8.3.1	Análisis de outliers . . . . .	196
8.3.1.1	Objetivo . . . . .	196
8.3.1.2	Inicialización . . . . .	196
8.3.1.3	Método . . . . .	196
8.3.1.4	Resultados . . . . .	197
8.3.1.4.1	Análisis de potencia . . . . .	197
8.3.1.4.1.1	Error Tipo I . . . . .	197
8.3.1.4.1.2	Efecto sobre la media . . . . .	197
8.3.1.4.1.3	Efecto sobre la cola . . . . .	197
8.3.1.4.2	Análisis de residuos . . . . .	197
8.3.1.4.2.1	Efecto sobre la media . . . . .	197
8.3.1.4.2.2	Efecto sobre la cola . . . . .	198
<b>9</b>	<b>Recapitulación de conclusiones y posibles vías futuras de trabajo</b>	<b>199</b>
9.1	Análisis de Supervivencia . . . . .	199
9.2	Redes Neuronales Autoorganizadas . . . . .	199
9.3	Tratamiento de outliers . . . . .	200
<b>V</b>	<b>Final</b>	<b>203</b>
<b>10</b>	<b>Agradecimientos</b>	<b>205</b>
	<b>Bibliografía</b>	<b>207</b>





# Índice de figuras

2.1	Distribución Gamma . . . . .	13
2.2	Distribución t-student . . . . .	13
2.3	Distribución Weibull . . . . .	14
2.4	Distribución Exponencial . . . . .	15
2.5	Distribución Gauss . . . . .	15
2.6	Distribución Ex-Gaussiana . . . . .	16
2.7	Distribución Ex-Gaussiana . . . . .	16
2.8	Se muestra aquí la (Fig. 5) referenciada en el artículo original. Puede verse como el punto de divergencia calculado no parece coincidir con el punto de divergencia gráfico real. Esto es lo que motiva las simulaciones siguientes. . . . .	19
2.9	Poca asimetría positiva. Efecto 20 ms. sobre la media . . . . .	20
2.10	Poca asimetría positiva. Efecto 40 ms. sobre la media . . . . .	21
2.11	Poca asimetría positiva. Efecto 80 ms. sobre la media . . . . .	21
2.12	Mucha asimetría positiva. Sin efecto . . . . .	22
2.13	Mucha asimetría positiva. Sin efecto. Falsos positivos. . . . .	23
2.14	Mucha asimetría positiva. Efecto 20 ms. sobre la media. . . . .	23
2.15	Mucha asimetría positiva. Efecto 40 ms. sobre la media. . . . .	24
2.16	Mucha asimetría positiva. Efecto 80 ms. sobre la media. . . . .	24
2.17	Análisis de potencia. Efectos sobre la media. . . . .	25
2.18	Mucha asimetría positiva. Efecto 20 ms. sobre la cola. . . . .	26
2.19	Mucha asimetría positiva. Efecto 40 ms. sobre la cola. . . . .	26
2.20	Mucha asimetría positiva. Efecto 80 ms. sobre la cola. . . . .	27
2.21	Análisis de potencia. Efectos sobre la cola. . . . .	27

3.1	Example of a divergence point. Cumulative distribution functions (CDFs) of observed latencies for two conditions that are identical until the marked point. Thereafter, two functions diverge. This representation is typical of plots from Reingold and Sheridan (e.g., Figure 2 of Reingold & Sheridan, 2018)	31
3.2	Density functions and cumulative distribution functions for ex-Gaussian distributions that have trivial divergence points. For both distributions $\mu = 300$ and $\sigma = 100$ , $\tau$ is 150 for the distribution is represented with <i>black lines</i> , and $\tau = 300$ for the distribution represented with a <i>light line</i>	34
3.3	Four scenarios for plausible mechanisms of decision-making under a simplified linear activation function. For each of the panels, the <i>bottom part</i> represents the activation processes, the <i>middle part</i> represent the density function for the latencies, and the <i>top panel</i> represents the cumulative density functions. In the four panels, the distributions that have trivial divergence points. See the main text for an explanation of each of the mechanisms	35
3.4	Density function and the cumulative distribution function for distributions that have a non-trivial divergence point. See main text for an explanation of each of the mechanisms	37
3.5	Different panels display the proportion of found divergence points assuming that there is no difference between the two conditions under comparison. The columns of panels represent the values of the $\mu$ parameter, and the rows represent the value of the $\tau$ parameter; $\sigma = 60$ for all simulations. As would be expected, the rate of spurious divergence points is determined by the number of trials per condition (the x-axis within each panel), and the number of participants (the lines within each panel)	41
3.6	Effects on $\mu$ and on $\tau$ taken from Reingold and Sheridan (2018) Appendix 1. These were parameters used to find non-trivial divergence points in their simulations. While this would represent an interesting set of parameters in their own right, they are somewhat unusual; this is because across many experiments, the effects on $\mu$ and on $\tau$ tend to be positively correlated, not negatively correlated, as in the figure. The <i>shade of the points</i> relates to the location of the divergence point according to the DPA method	42

3.7	We present two examples of divergence points. Panels a and b show crossovers in the survival functions; the fast condition is shown with dashed lines . . . . .	43
3.8	Illustration of how small changes can yield large fluctuations in the DPA results. The three panels show distributions with small variations in the $\mu$ parameter (150ms. to 160ms. to 161ms.) for one of the conditions, which yields widely different divergence points . . . . .	44
4.1	Aprendizaje de letras. Similitud baja. . . . .	75
4.2	Aprendizaje de letras. Similitud moderada. . . . .	76
4.3	Aprendizaje de letras. Alta similitud. . . . .	76
4.4	Método 1. Mapa de letras entrenado. . . . .	77
4.5	Aprendizaje de palabras. Comparación de representaciones. Poca similitud. . . . .	80
4.6	Aprendizaje de palabras. Comparación de representaciones. Similitud moderada. . . . .	80
4.7	Aprendizaje de palabras. Comparación de representaciones. Alta similitud. . . . .	81
4.8	Método 1. Mapa de palabras entrenado. . . . .	81
4.9	Aprendizaje de letras. Similitud baja. . . . .	82
4.10	Aprendizaje de letras. Similitud moderada. . . . .	82
4.11	Aprendizaje de letras. Alta similitud. . . . .	83
4.12	Método 2. Mapa de letras entrenado. . . . .	84
4.13	Aprendizaje de palabras. Comparación de representaciones. Poca similitud. . . . .	84
4.14	Aprendizaje de palabras. Comparación de representaciones. Similitud moderada. . . . .	85
4.15	Aprendizaje de palabras. Comparación de representaciones. Alta similitud. . . . .	85
4.16	Método 2. Mapa de palabras entrenado. . . . .	86
4.17	Aprendizaje de letras. Similitud baja. . . . .	87
4.18	Aprendizaje de letras. Similitud moderada. . . . .	87
4.19	Aprendizaje de letras. Alta similitud. . . . .	88
4.20	Método 3. Mapa de letras entrenado. . . . .	89
4.21	Aprendizaje de palabras. Comparación de representaciones. Poca similitud. . . . .	89

4.22	Aprendizaje de palabras. Comparación de representaciones. Similitud moderada . . . . .	90
4.23	Aprendizaje de palabras. Comparación de representaciones. Alta similitud. . . . .	90
4.24	Método 3. Mapa de palabras entrenado. . . . .	91
4.25	Aprendizaje de letras. Error de Precisión de la Proyección. Método 1. . . . .	92
4.26	Aprendizaje de letras. Error de Precisión de la Proyección. Método 2. . . . .	92
4.27	Aprendizaje de letras. Error de Precisión de la Proyección. Método 3. . . . .	93
4.28	Aprendizaje de letras. Error de Precisión de la Proyección. Método 1. . . . .	93
4.29	Aprendizaje de letras. Error de Precisión de la Proyección. Método 2. . . . .	94
4.30	Aprendizaje de letras. Error de Precisión de la Proyección. Método 3. . . . .	94
5.1	Transformaciones de datos . . . . .	101
5.2	Transformaciones de datos. Función de densidad . . . . .	102
5.3	Distribución ExGauss de 10 ítems . . . . .	104
5.4	Distribución de los datos en un supersujeto. Efectos sobre la media. Sin outliers . . . . .	106
5.5	Distribución de los datos en un supersujeto. Efectos sobre la media. 10 % outliers . . . . .	107
5.6	Distribución de los datos en un supersujeto. Efectos sobre la cola. Sin outliers . . . . .	108
5.7	Distribución de los datos en un supersujeto. Efectos sobre la cola 10 % outliers . . . . .	109
5.8	Dispersión de los datos en un supersujeto. Efectos sobre la media . . . . .	113
5.9	Dispersión de los datos en un supersujeto. Efectos sobre la cola . . . . .	114
6.1	Distribución ExGauss de 10 ítems . . . . .	118
6.2	Análisis de falsos positivos . . . . .	120
6.3	Efecto de 20 ms. sobre la media. . . . .	121
6.4	Efecto de 50 ms. sobre la media . . . . .	121
6.5	Efecto de 100 ms. sobre la media . . . . .	122

6.6	Efecto de 20 ms. sobre la cola . . . . .	123
6.7	Efecto de 50 ms. sobre la cola . . . . .	123
6.8	Efecto de 100 ms. sobre la cola . . . . .	124
6.9	Sin efectos ni variabilidad intersujetos. QQ-Norm. . . . .	133
6.10	Sin efectos ni variabilidad intersujetos. Histograma de suma de resíduos . . . . .	134
6.11	Sin efectos ni variabilidad intersujetos. Histograma. . . . .	135
6.12	Efecto sobre la media 20 ms. QQ-Norm. . . . .	136
6.13	Efecto sobre la media 20 ms. Histograma de suma de resíduos . .	136
6.14	Efecto sobre la media 20 ms. Histograma. . . . .	137
6.15	Efecto sobre la media 50 ms. QQ-Norm. . . . .	138
6.16	Efecto sobre la media 50 ms. QQ-Norm. . . . .	138
6.17	Efecto sobre la media 50 ms. Histograma. . . . .	139
6.18	Efecto sobre la media 100 ms. QQ-Norm. . . . .	139
6.19	Efecto sobre la media 100 ms. Histograma de suma de resíduos .	140
6.20	Efecto sobre la media 100 ms. Histograma. . . . .	141
6.21	Efecto sobre la cola 20 ms. QQ-Norm. . . . .	141
6.22	Efecto sobre la cola 20 ms. Histograma de suma de resíduos. . .	142
6.23	Efecto sobre la cola 20 ms. Histograma. . . . .	143
6.24	Efecto sobre la cola 50 ms. QQ-Norm. . . . .	144
6.25	Efecto sobre la cola 50 ms. Histograma de suma de resíduos. . .	144
6.26	Efecto sobre la cola 50 ms. Histograma. . . . .	145
6.27	Efecto sobre la cola 100 ms. QQ-Norm. . . . .	146
6.28	Efecto sobre la cola 100 ms. Histograma de suma de resíduos. .	147
6.29	Efecto sobre la cola 100 ms. Histograma. . . . .	148
7.1	Distribución ExGauss de 10 items . . . . .	154
7.2	Análisis de falsos positivos . . . . .	160
7.3	Efecto de 20 ms. sobre la media. . . . .	161
7.4	Efecto de 50 ms. sobre la media . . . . .	162
7.5	Efecto de 100 ms. sobre la media . . . . .	162
7.6	Efecto de 20 ms. sobre la cola . . . . .	163
7.7	Efecto de 50 ms. sobre la cola . . . . .	164
7.8	Efecto de 100 ms. sobre la cola . . . . .	164
7.9	Sin efectos ni variabilidad intersujetos. QQ-Norm. . . . .	173
7.10	Sin efectos ni variabilidad intersujetos. Histograma de suma de resíduos . . . . .	174

7.11 Sin efectos ni variabilidad intersujetos. Histograma. . . . .	175
7.12 Efecto sobre la media 10 ms. QQ-Norm. . . . .	176
7.13 Efecto sobre la media 10 ms. Histograma de suma de residuos . .	176
7.14 Sin efectos ni variabilidad intersujetos. Histograma. . . . .	177
7.15 Efecto sobre la media 20 ms. QQ-Norm. . . . .	178
7.16 Efecto sobre la media 20 ms. QQ-Norm. . . . .	179
7.17 Efecto sobre la media 20 ms. Histograma. . . . .	180
7.18 Efecto sobre la media 40 ms. QQ-Norm. . . . .	181
7.19 Efecto sobre la media 40 ms. Histograma de suma de residuos . .	181
7.20 Efecto sobre la media 40 ms. Histograma. . . . .	182
7.21 Efecto sobre la cola 20 ms. QQ-Norm. . . . .	183
7.22 Efecto sobre la cola 20 ms. Histograma de suma de residuos. . .	183
7.23 Efecto sobre la cola 20 ms. Histograma. . . . .	184
7.24 Efecto sobre la cola 50 ms. QQ-Norm. . . . .	185
7.25 Efecto sobre la cola 50 ms. Histograma de suma de residuos. . .	185
7.26 Efecto sobre la cola 50 ms. Histograma. . . . .	186
7.27 Efecto sobre la cola 100 ms. QQ-Norm. . . . .	187
7.28 Efecto sobre la cola 100 ms. Histograma de suma de residuos. . .	187
7.29 Efecto sobre la cola 100 ms. Histograma. . . . .	188

# Índice de cuadros

3.1	Rate of spurious divergence points for simulations with null effects. The latencies were generated with parameters $\mu = 100$ , $\sigma = 60$ , and $\tau = 50$ ms. . . . .	39
4.1	Letras. Estímulos de entrenamiento. . . . .	52
4.2	Palabras de tres letras. Estímulos de entrenamiento. . . . .	53



## Parte I

# Presentación



# Capítulo 1

## Introducción

La siguiente tesis doctoral se presenta como un trabajo de investigación relacionado con el análisis estadístico de tiempos de respuesta. El objeto de estudio de este trabajo pretende aportar algo de información sobre el tratamiento de *outliers* incorporando la alternativa de transformación  $\frac{1}{t^2}$ . Para el análisis de fiabilidad de las transformaciones de datos outliers se suele utilizar el *modelo lineal t-test* que busca un patrón lineal en el conjunto unificado de datos; como novedad en el presente trabajo, se usará un *modelo generalizado lineal mixto* que pretende mejorar el modelo lineal primigenio diferenciando los datos en su origen (sujeto e ítem). En todos los casos los tiempos de respuesta se consideran medidos en milisegundos.

El estudio está realizado dentro del marco de la psicología / neurociencia cognitiva y el reconocimiento de palabras. Está dividido en dos bloques: *I)* En el primer bloque se realizó una fase preparatoria de la investigación donde se profundizó sobre los métodos computacionales existentes relacionados con el aprendizaje lector y análisis de tiempos de respuesta de reconocimiento de palabras. *II)* Una vez completada esta fase de inmersión previa, damos paso al segundo bloque de la investigación donde nos adentraremos dentro del desarrollo específico del trabajo. En este segundo bloque se introducirá el trabajo con los *outliers* o *datos extremos*; para esto, se analizará la potencia específica de la transformación  $\frac{1}{t^2}$  estudiada tanto desde el punto de vista de un *modelo lineal* como de un *modelo lineal generalizado mixto*.

Antes de adentrarnos en las diferentes partes que componen los dos bloques

citados anteriormente, comenzaremos describiendo las diferentes fases que componen la investigación para enmarcar el trabajo y establecer un primer contacto con la terminología asociada.

En un primer momento se estudió el trabajo de Reingold, Reichle, Glaholt & Sheridan (2012). Los conceptos clave de este estudio son los de curva de supervivencia, *bootstrap* y punto de divergencia. Dado un conjunto de datos o lecturas temporales, siempre dentro del reconocimiento de palabras,  $t_1, t_2, \dots, t_n$  se calcula la *Supervivencia* de un valor concreto  $t_0$ , como el número de lecturas con valor superior al dado,  $S(t_0) = |t_j : j = 1, \dots, n, t_j \geq t_0|$ . Partiendo del mismo conjunto de datos,  $t_1, t_2, \dots, t_n$ , la técnica *bootstrap* se basa en generar  $k$  muestras a partir del conjunto de lecturas base mediante un muestreo con reemplazamiento. Pueden consultarse las bases teóricas de esta técnica en Efron (1979) y Efron (1987). Si en lugar de un conjunto de datos único, consideramos dos conjuntos de datos, procedentes de tomar la información de dos condiciones de lectura separadas (por ejemplo, palabras de baja frecuencia frente a palabras de alta frecuencia) se generarían dos curvas de supervivencia diferenciadas. El objetivo de esta manipulación es obtener el punto de divergencia  $t_0$  en el cual ambas curvas se diferencian.

El segundo método preparatorio al resto del trabajo se basó en tratar de examinar críticamente el procedimiento expuesto en Polk, Lacey, Nelson, Demiralp, Newman, Kraus, Raheja & Farah (2009). En este artículo, el autor se basa en una red neuronal autoorganizada del tipo de las presentadas en Kohonen (1990) para modelar el proceso de aprendizaje lector. Los puntos de partida son: *a*) un conjunto de neuronas organizado en una cuadrícula  $m \times n$ , cada una de las cuales tiene un peso de  $k$  componentes aleatorias, y *b*) un abecedario o conjunto de estímulos que formalizaremos como un conjunto de vectores de pesos aleatorios y fijos de  $k$  componentes. En cada iteración del método se toma uno de los estímulos de forma aleatoria y se determina la neurona más cercana en peso al estímulo presentado. Esta será la neurona activa. Las restantes neuronas se ajustarán en peso en base a su distancia a la neurona activa y similitud con el estímulo.

Una vez completados los estudios previos a modo de preparación, se llega a la parte correspondiente a la aportación específica del trabajo. En este bloque se tratan los diferentes métodos de tratamiento de *outliers*. Según Barnett & Lewis (1994) se define un *outlier* o dato extremo como “aquella observación (o

conjunto de observaciones) inconsistente con el resto del conjunto de datos”. La presencia de datos extremos en un conjunto de datos hace que las conclusiones estadísticas se puedan ver afectadas y dar paso a conclusiones no realistas. Son diferentes las técnicas que se han empleado a lo largo de los años para reducir el impacto de esta clase de datos. A continuación facilitamos una muestra de las mismas:

- Truncamiento de la muestra. Nos quedaremos con los datos dentro del rango  $\bar{X} \pm 1,5\tilde{X}$ ,  $\bar{X} \pm 2\tilde{X}$ , etc.
- Uso de alternativas a la media. Tener en cuenta que la media es un valor que se ve notablemente afectado por la presencia de datos extremos. Pueden emplearse como alternativas la *media recortada* / *trimming*, que elimina un porcentaje de datos de su cálculo, o la *media winsorizada*, que reemplaza un porcentaje de los datos por valores menos extremos.
- Transformaciones de datos. Este será el tipo de tratamiento que se utilizará en el presente estudio y que se basa en diferentes funciones de preprocesamiento sobre las lecturas.

Centrándonos ahora en la tercera opción correspondiente a las transformaciones de datos, se parte del desarrollo preexistente de Ratcliff (1993) en el que se estudió la efectividad de los métodos clásicos  $t$ ,  $\log(t)$  y  $\frac{1}{t}$  mediante un tratamiento de modelo lineal. A estas opciones de transformación se incorpora la posibilidad  $\frac{1}{t^2}$ . Se estudia a continuación la validez del mismo tipo de transformaciones de datos pero ya no en base a un modelo lineal sino a un modelo lineal generalizado mixto. Este último tipo de modelo se introdujo inicialmente en el texto de Nelder & Wedderburn (1972). Si el modelo lineal trata de buscar un patrón único para el conjunto global de datos, el modelo lineal generalizado mixto tiene en cuenta que los datos pueden presentar variabilidad tanto por sujeto como por ítem y condición.

La generación de datos de todos los apartados del trabajo se hace mediante muestras *Ex-Gauss*, es decir, como suma de una distribución *Exponencial*  $Ex(\tau)$ , y una *Normal* o *Gauss*  $N(\mu, \sigma)$ .

Las técnicas de estudio se analizarán utilizando datos referentes a tiempos de respuesta de reconocimiento de palabras. Esta técnica está estrechamente ligada al área de trabajo del *Departamento de Metodología de las Ciencias del Comportamiento* de la *Universitat de València*.

A continuación, se ofrece una panorámica de las diferentes partes de las que se

compone esta tesis doctoral que nos permita entender el desarrollo del trabajo, y con este objetivo, se desarrollan dos bloques de conocimiento.

Veamos en primer lugar la organización del trabajo:

- Bloque 1
  - **Parte I (Survival)**. En Reingold et al. (2012) se propone una técnica de cálculo del punto de divergencia entre dos curvas de supervivencia correspondientes dos condiciones diferentes basada en la técnica *bootstrap* aplicada sobre curvas de supervivencia. Esta parte se organiza en dos secciones: *a)* Simulaciones computacionales para valorar la fiabilidad de la técnica, y *b)* Refutación formal de la técnica de Reingold et al. (2012) en Gomez, Breithaupt, Perea & Rouder (2021).
  - **Parte II (SOM)**. En Polk et al. (2009) se planteó la aplicación del modelo de redes autoorganizadas de Kohonen (1990) (*Self Organizing Map*) como una opción de aprendizaje lector. Se parte de un mapa de pesos aleatorio y con una activación basada en la distribución Normal a partir de la neurona activa frente a cada estímulo. Al final del aprendizaje, cada par de letras y/o palabras, ya sean presentadas en minúscula y/o en mayúscula, generan el mismo patrón neuronal de activación.
- Bloque 2
  - **Parte III (Outliers - análisis de datos y transformaciones)**. En los siguientes apartados se estudian diferentes transformaciones de datos a modo de técnicas para el tratamiento de datos outliers. Se plantean aquí las diferentes transformaciones posibles y la justificación teórica de su empleo.
  - **Parte IV (Outliers - modelo lineal)**. Las diferentes técnicas de análisis de tratamiento de datos outliers son analizadas desde la perspectiva de un modelo lineal de forma similar al trabajo de Ratcliff (1993). Se incorpora el cuadrado de la distribución inversa  $\frac{1}{t^2}$  como alternativa a las ya existentes y se estudia su bondad frente a las técnicas preexistentes.
  - **Parte V (Outliers - modelo lineal generalizado mixto)**. Se reproduce el mismo tipo de análisis que en la parte III pero mediante el empleo del análisis de modelos lineales generalizados mixtos.

En la **Parte I** se pretendió un primer acercamiento a los métodos estadísti-

cos computacionales, tomando como punto de partida un trabajo reciente que planteaba una estrategia innovadora. Sin embargo, en el desarrollo de las simulaciones computacionales correspondientes, se obtuvo que la que se planteaba como una técnica innovadora y prometedora resultó ser poco estable. Se plantearon formalmente los resultados y argumentos para su refutación en Gomez et al. (2021). Hubiera sido deseable obtener unos resultados que validaran este método para poderlo confrontar con datos reales.

El siguiente desarrollo se basó en lo que finalmente se ha organizado en la **Parte II**. Los resultados fueron similares a los originales pero no se dispuso del código original detallado. Cuando se intentó contactar con el autor del trabajo original Polk et al. (2009) y con la editorial correspondiente para confirmar que nuestro algoritmo coincidía con el desarrollo algorítmico original, no se obtuvo respuesta en los diferentes intentos. Por este motivo, al carecer de retroalimentación con los responsables de la publicación original y no poder contrastarse ambos desarrollos, no se pudo profundizar en las posibilidades de este planteamiento y esta parte de la investigación debió quedar como una réplica sin posibilidad de contraste. Aun sin disponer de retroalimentación en este punto para confirmar nuestros resultados, el amplio uso de las técnicas de redes neuronales en neurociencia y el respaldo constante en los trabajos de Kohonen (1990) y Kangas, Kohonen & Laaksonen (1990) motivan la incorporación del desarrollo de esta parte en el compendio final.

Quedan finalmente **Parte I** y **Parte II** como un testimonio del trabajo realizado, aunque no como el núcleo principal del presente desarrollo. Las partes **Parte III**, **Parte IV** y **Parte V** son finalmente las partes en las que se ha aportado una novedad consistente partiendo de unos trabajos previos.





## Parte II

# Introducción a los métodos computacionales de tratamiento de datos en tiempos de respuesta de reconocimiento de palabras



## Capítulo 2

# Curvas de supervivencia promedio y punto de divergencia mediante bootstrap e análisis de tiempos de respuesta de reconocimiento de palabras

### 2.1 Introducción

El análisis de curvas de supervivencia ha sido una metodología de trabajo ampliamente utilizada en Psicología y otras disciplinas. En el trabajo de Reingold, Reichle, Mackenzie & Sheridan (2012) se propone un método que combina las curvas de supervivencia con la técnica *bootstrap* para estimar el punto de divergencia cuando comparamos los tiempos de respuesta (en adelante *TR*) de una misma población sometida a dos condiciones diferentes. Los resultados ofrecidos por los autores del trabajo original resultaban esperanzadores y el presente

trabajo de investigación se inició para testear las bondades y puntos de mejora de la nueva técnica. Sin embargo, las simulaciones computacionales y consiguientes análisis que se llevaron a cabo a tal efecto llevaron al equipo responsable del presente trabajo a pensar que tal vez la técnica no fuera consistente. En todo caso, este desarrollo fue un punto de partida válido para un acercamiento a los métodos de análisis computacional de datos en Psicología y Neurolingüística. En el desarrollo que sigue se plantean diferentes simulaciones de datos y se analizan los resultados derivados de aplicar la técnica propuesta por Reingold et al. (2012). En base a estas primeras simulaciones y de forma complementaria a las mismas, en el trabajo de Gómez, Breithaupt, Perea & Rouder (2021), se ha justificado la falta de rigor teórico del innovador método. El desarrollo de Gomez et al. (2021) se transcribe en un apartado posterior quedando fuera de esta sección. Esta parte del trabajo es transcripción directa del artículo publicado por lo que en lugar de estar en Español como el resto de este desarrollo lo está en Inglés. Tener en cuenta que el que este apartado del trabajo sea una transcripción directa del artículo original también justifica el que la bibliografía correspondiente se inserte al final del mismo y no al final del trabajo global.

## 2.2 Resumen del método teórico

Los datos de partida pretenden corresponder a tiempos de respuesta de reconocimiento de palabras tal y como corresponde al ámbito de la Psicología Cognitiva. En un estudio basado en datos reales, como es frecuente en este campo, los datos procederían de técnicas de tipo *ERPs* o *Potenciales Evocados por Eventos*, en que registra la respuesta electrofisiológica o los patrones de activación neuronal frente a diferentes estímulos presentados al sujeto. Pueden revisarse las bases técnicas de esta técnica de campo en Luck (2005) y Luck & Kappenman (2012). En el caso que nos ocupa, para poder evaluar la consistencia de una técnica particular, recurriremos a la simulación computacional de datos, que pretenden emular *TR's* de reconocimiento de palabras. Los *TR's* frente a reconocimiento de palabras acostumbra a ser valores positivos con menor frecuencia en rangos altos, es decir, con distribución asimétrica positiva. Por este motivo, podríamos encontrarlos generados por distribuciones estadísticas de tipo *Gamma*, *Wald* o *Weibull*.

Sin embargo, lo más frecuente suele ser emplear una distribución *Ex-*

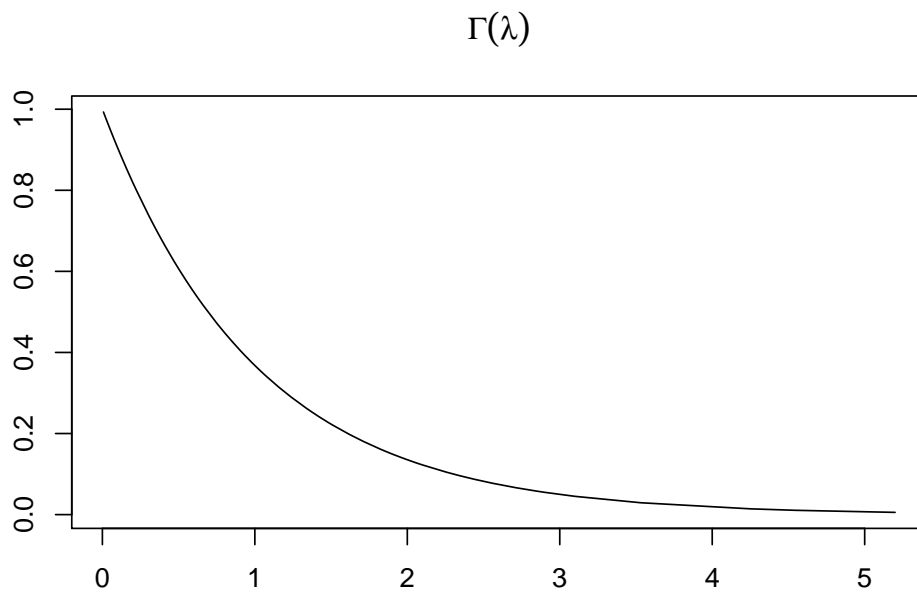


Figura 2.1: Distribución Gamma

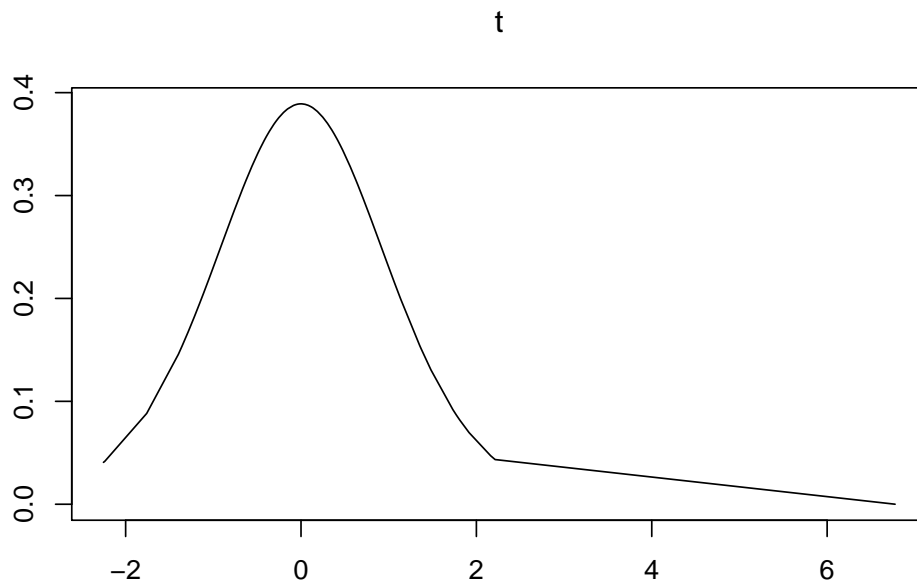


Figura 2.2: Distribución t-student

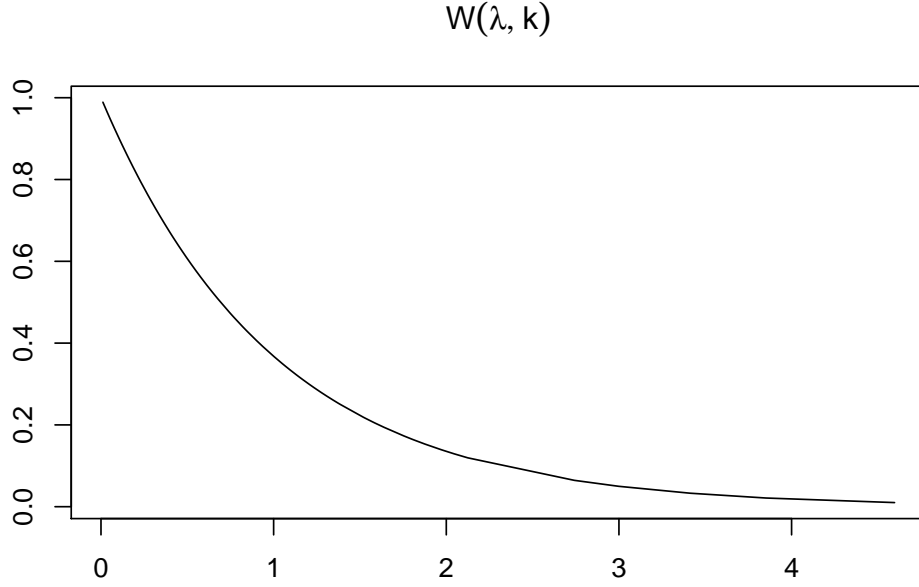


Figura 2.3: Distribución Weibull

*Gaussiana* (suma de una distribución *Exponencial* y una *Normal* o *Gauss*), como puede leerse en Balota & Spieler (1999), Heathcote, Popiel & Spieler (1991) y Ratcliff & Murdock (1976).

Para entender el comportamiento de la distribución *Ex-Gaussiana* veamos algunos ejemplos sobre el impacto gráfico en la distribución al variar alguno de los parámetros  $\mu$ ,  $\sigma$  y/o  $\tau$ .

Dado un *TR*  $t_0$  válido, la *Supervivencia* en dicho valor se define como

$$S(t_0) = P(TR > t_0)$$

. Frente a un conjunto de  $n$  datos o lecturas  $t_1, t_2, \dots, t_n$ , se calcula la supervivencia de  $t_0$  (expresada en %) como

$$S(t_0) = \frac{|\{t_j/j \in \{1, \dots, n\}, t_j \geq t_0\}|}{n} \cdot 100 \%$$

, es decir, el porcentaje o tasa porcentual de lecturas que superan el valor  $t_0$  dado.

En las siguientes líneas se procede a presentar el objeto de trabajo y los

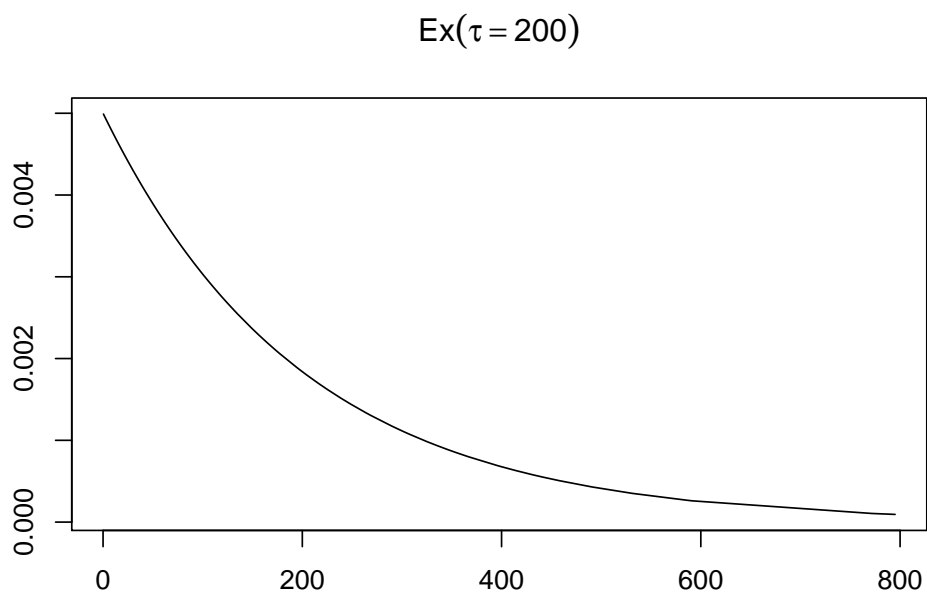


Figura 2.4: Distribución Exponencial

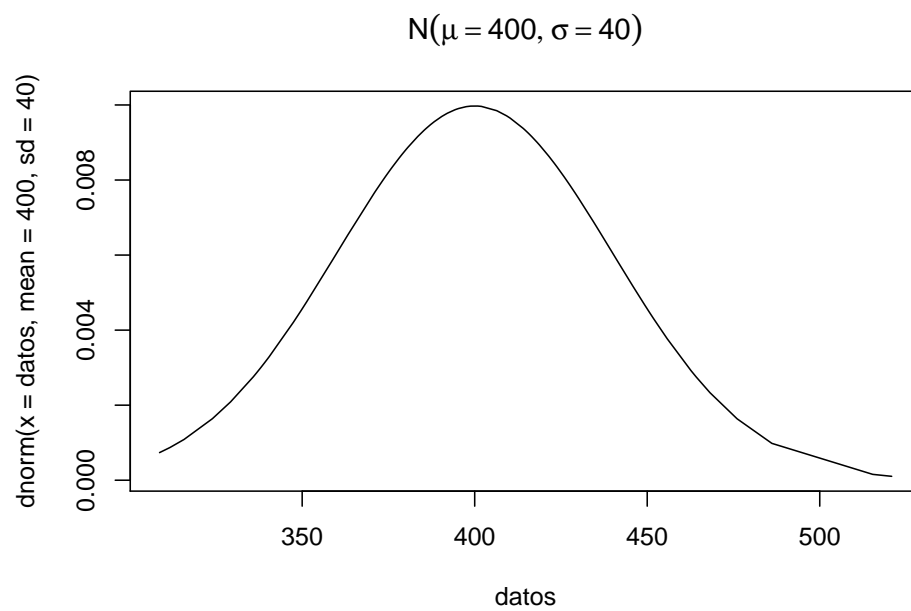


Figura 2.5: Distribución Gauss

$$\text{Ex}(\tau = 200) + \text{N}(\mu = 400, \sigma = 40) = \text{Ex - Gauss}(\mu = 400, \sigma = 40, \tau = 200)$$

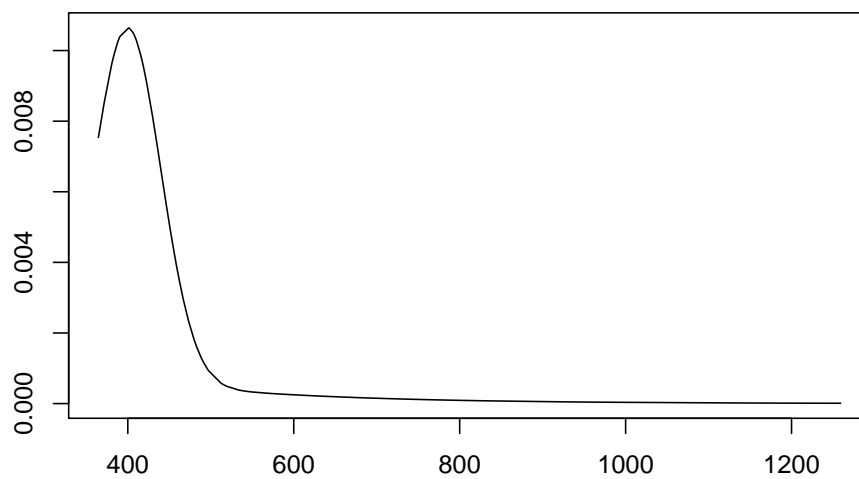


Figura 2.6: Distribución Ex-Gaussiana

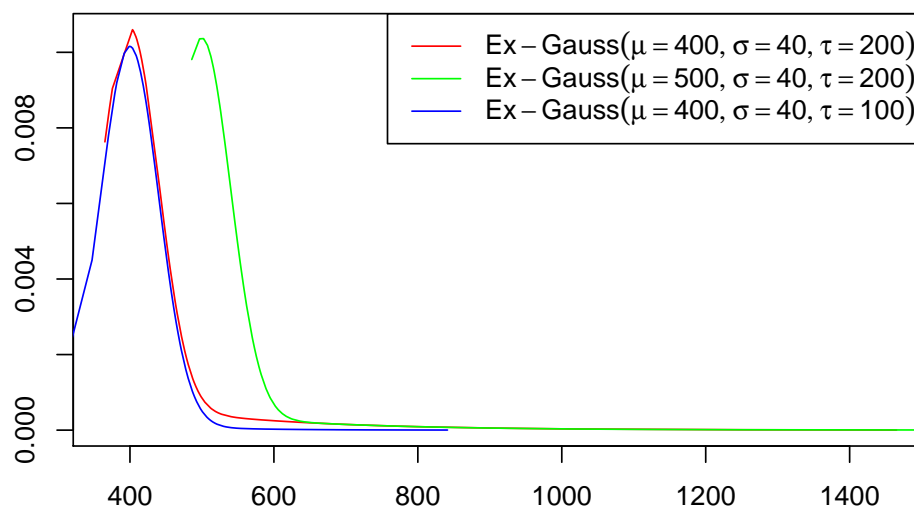


Figura 2.7: Distribución Ex-Gaussiana



parámetros definitorios con la misma notación que en el código que le sigue.

Sean  $n_S$  sujetos, 2 condiciones y  $n_I$  items por sujeto y condición. Entenderemos los  $TR$  simulados mediante distribución Ex-Gaussiana. Mentalmente podemos entender los datos estructurados como 2 matrices de  $n_S \times n_I$  lecturas.

Entendamos lo siguiente como un breve recordatorio del método propuesto de las explicaciones técnicas más detalladas en el trabajo de Reingold et al. (2012).

- Dados  $n_I$  items o lecturas (en un sujeto particular), entendidos en  $ms$ . con valores enteros positivos y ordenados de forma creciente por simplicidad, independientemente de la condición, podemos calcular la supervivencia de cada  $TR$  dado. Los valores de supervivencia correspondientes a  $TR$  no presentes en la muestra pero sí dentro del rango de la misma se calculan por interpolación lineal entre los dos valores de supervivencia de los tiempos de respuesta más cercanos sí presentes en la muestra.
- Cuando se generan las  $n_S$  curvas de supervivencia individuales (una por sujeto) se puede generar una curva de supervivencia promedio de las calculadas. Cada valor de dicha curva de supervivencia promedio es la media de los correspondientes valores de supervivencia en cada curva o sujeto.
- Se repite el proceso de calcular la curva de supervivencia promedio en cada una de las dos condiciones, con lo que obtenemos dos curvas de supervivencia de esta clase.
- En el momento en que quedan definidas las dos curvas de supervivencia promedio se calcula el punto de divergencia como el primer  $TR$  en el que se aprecia una diferencia notable entre ambas curvas.
- Una vez aplicada la técnica de *bootstrap* (remuestreo con reemplazamiento) una cantidad suficiente de veces se tiene en cuenta el intervalo de confianza de diferencia entre las curvas en cada  $TR$ . Pueden repasarse los fundamentos de esta técnica en Efron (1979) y Efron (1987).

También podemos acudir a la fuente del trabajo original de Reingold et al. (2012):

*Next, we computed survival curves for first fixation durations in each frequency by preview condition. For each 1-ms time bin  $t$  ( $t$  was varied from 0 to 600 ms), the percentage of first fixations with a duration greater than  $t$  constituted the percent survival at time  $t$ . The survival curve for each preview by frequency condition was computed separately for each participant, and then averaged across participants. As can be seen by an inspection of Fig. 5 (Panels e-f),*

in both the valid and the invalid preview conditions, the HF and LF survival curves appear to diverge. Importantly, this divergence point corresponds by definition to the shortest first fixation duration value at which word frequency had a significant impact. In order to estimate the divergence point between the HF and LF survival curves, we employed a bootstrap re-sampling procedure (Efron & Tibshirani, 1994). On each iteration of this procedure, the set of observations (first fixation durations) for each participant in each condition was randomly re-sampled with replacement. For each iteration of the bootstrap procedure, individual participant's survival curves were then computed and averaged. Next, the value for each 1-ms bin in the HF survival curve was subtracted from the corresponding value in the LF survival curve. This procedure was repeated 10,000 times, and the obtained differences for each bin were then sorted in order of magnitude. The range between the 5th and the 9,995th value was then defined as the confidence interval of the difference for each bin (given the multiple comparisons we performed, we used this conservative confidence interval in order to protect against making a Type I error). To compute the divergence point between the LF and HF survival curves, we identified the time bins for which LF survival rate was significantly greater than HF survival rate (i.e., for which the lower bound of the confidence interval of the difference between the LF and HF curves was greater than zero). The divergence point was then defined as the earliest significant difference point that was part of a run of five consecutive significant difference points (significant differences between the LF and HF curves are shown in Panels e–f of Fig. 5 as a row of asterisks above the survival curves).

## 2.3 Simulaciones computacionales

Los *TR* se simularán, como es frecuente, mediante distribuciones  $Ex - Gauss(\mu, \sigma, \tau)$ , tal y como se introdujo en el apartado anterior. Para las simulaciones se emplearán condiciones realistas, enfrentando dos supersujetos y variando tanto el número de sujetos como el número de items y el efecto sobre la media.

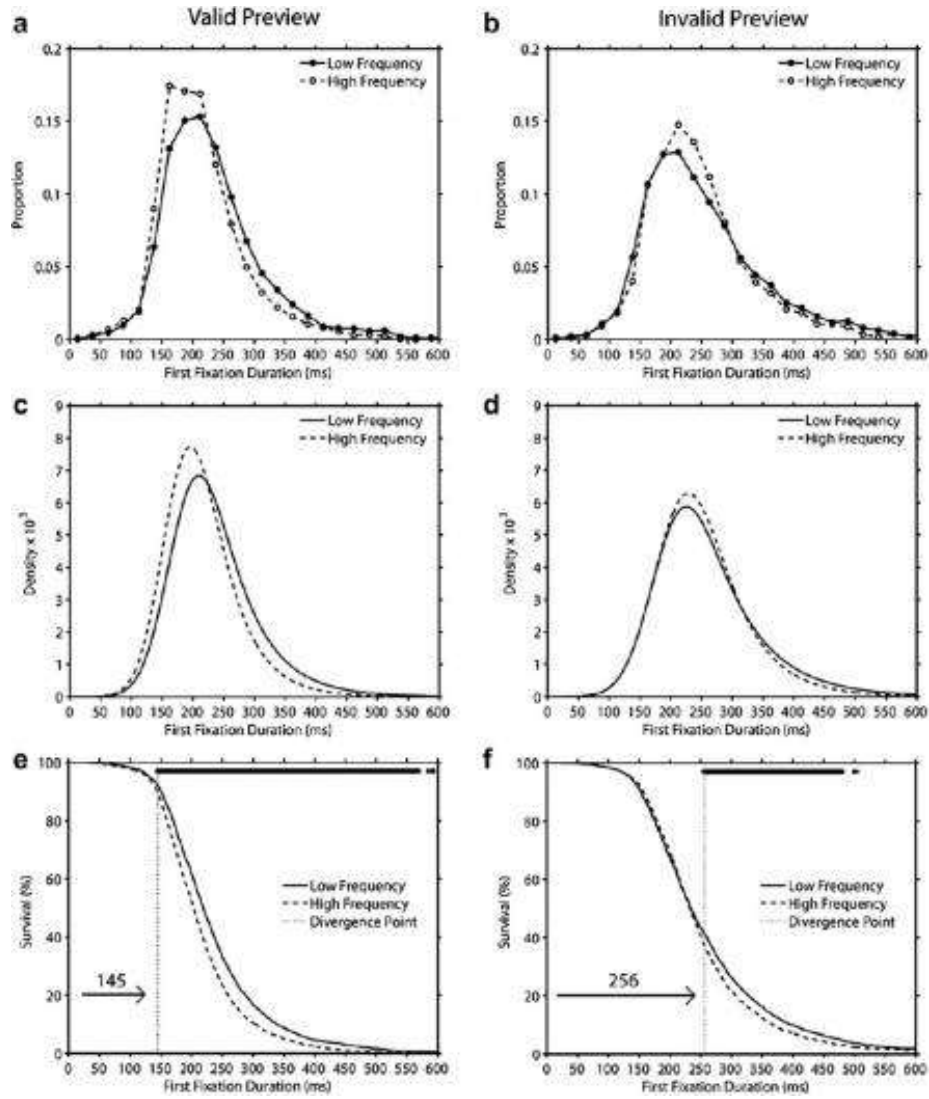


Figura 2.8: Se muestra aquí la (Fig. 5) referenciada en el artículo original. Puede verse como el punto de divergencia calculado no parece coincidir con el punto de divergencia gráfico real. Esto es lo que motiva las simulaciones siguientes.

### 2.3.1 Simulaciones realistas con poca asimetría positiva

La distribución base, en este caso será del tipo  $Ex - Gauss(\mu = 541, \sigma = 68, \tau = 20)$ , sobre la que aplicaremos efectos sobre la media ( $\mu$ ) de 20, 40 y 80 ms.

#### 2.3.1.1 Efecto sobre la media

$ExG(\mu = 541, \sigma = 68, \tau = 20)$  vs.  $ExG(\mu = 541 + 20, \sigma = 68, \tau = 20)$

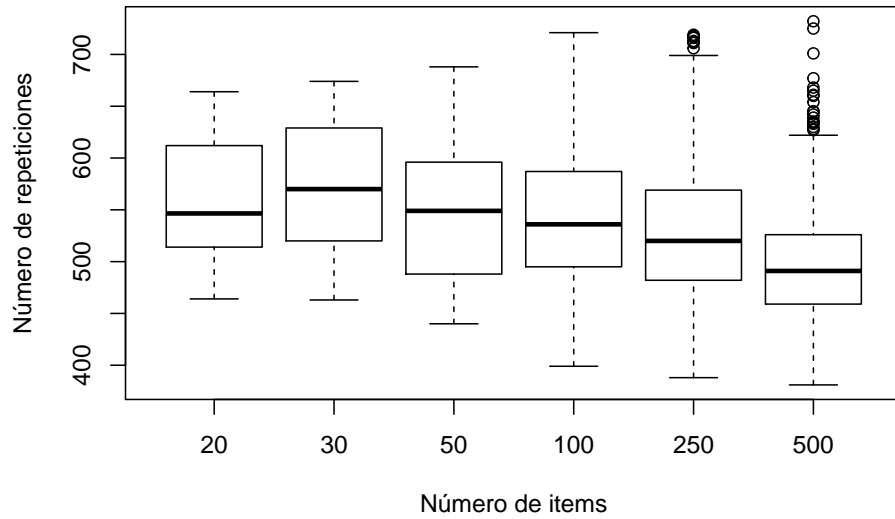


Figura 2.9: Poca asimetría positiva. Efecto 20 ms. sobre la media

Al aumentar el número de items disminuye la media a la par que se reduce el número de datos fuera del gráfico de cajas o outliers. Además, se incrementa el ritmo de descenso de la media al aumentar el número de items cuando vamos ampliando el efecto sobre la media.

### 2.3.2 Simulaciones realistas con mucha asimetría positiva

La distribución base, en este caso será del tipo  $ExGauss(\mu = 541, \sigma = 68, \tau = 115)$ , sobre la que aplicaremos efectos sobre la media ( $\mu$ ) de 20, 40 y 80 ms.

$\text{ExG}(\mu = 541, \sigma = 68, \tau = 20)$  vs.  $\text{ExG}(\mu = 541 + 40, \sigma = 68, \tau = 20)$

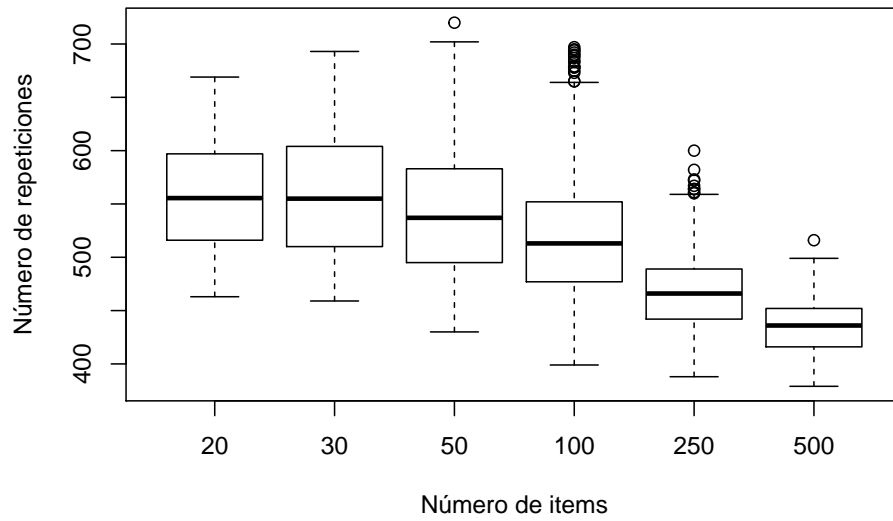


Figura 2.10: Poca asimetría positiva. Efecto 40 ms. sobre la media

$\text{ExG}(\mu = 541, \sigma = 68, \tau = 20)$  vs.  $\text{ExG}(\mu = 541 + 80, \sigma = 68, \tau = 20)$

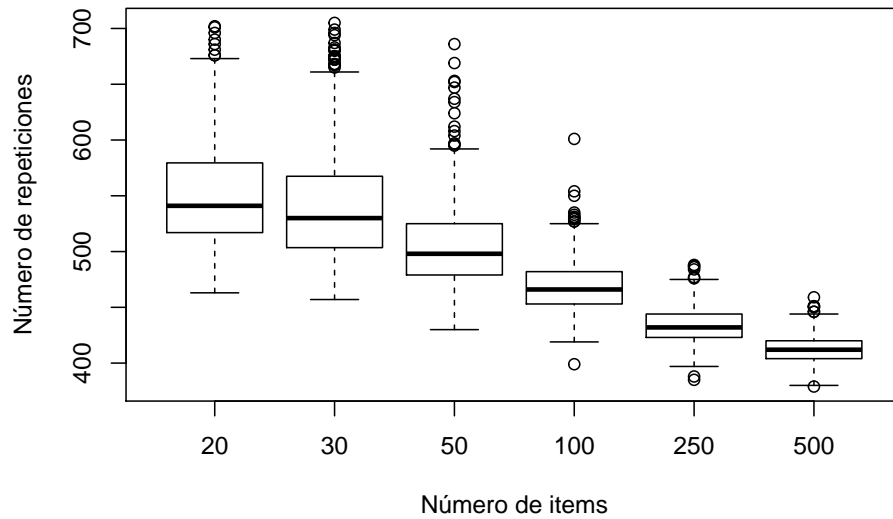


Figura 2.11: Poca asimetría positiva. Efecto 80 ms. sobre la media

### 2.3.2.1 Sin efecto

ExG( $\mu = 541, \sigma = 68, \tau = 115$ ) vs. ExG( $\mu = 541, \sigma = 68, \tau = 115$ ) (sin efecto)

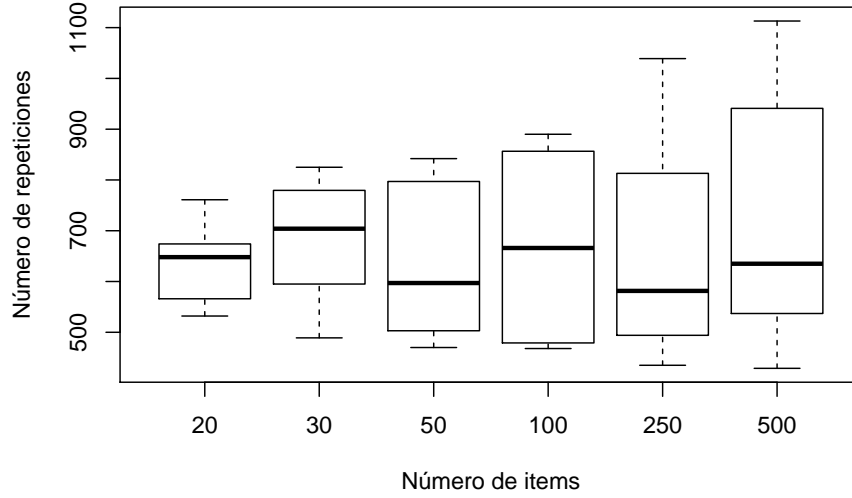


Figura 2.12: Mucha asimetría positiva. Sin efecto

Cuando no se aplica efecto sobre ninguno de los parámetros de la distribución ExGauss el número de falsos positivos se mantiene estable entre el 1 y el 2 %.

### 2.3.2.2 Efecto sobre la media

De forma similar a como vimos en las simulaciones con poca asimetría positiva, al aumentar el número de items disminuye la media a la par que se reduce el número de datos fuera del gráfico de cajas. También de forma similar al caso con poca asimetría positiva, un aumento en el efecto incrementa el ritmo de descenso de la media al ampliar el número de items.

### 2.3.2.3 Estudio de la mediana con efecto sobre la media

Según era previsible el punto de divergencia disminuye conforme se incrementa el tamaño del efecto sobre la media. Cabe comentar que la diferencia entre los casos correspondientes a los diferentes efectos no es estable al cambiar el tamaño del efecto. Manejando hasta 500 items y fijando un efecto sobre la media

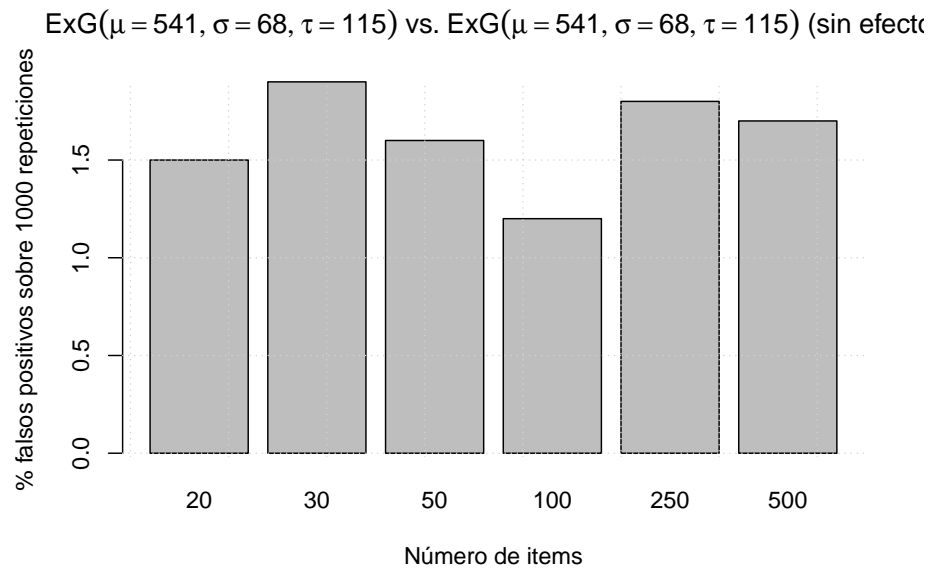


Figura 2.13: Mucha asimetría positiva. Sin efecto. Falsos positivos.

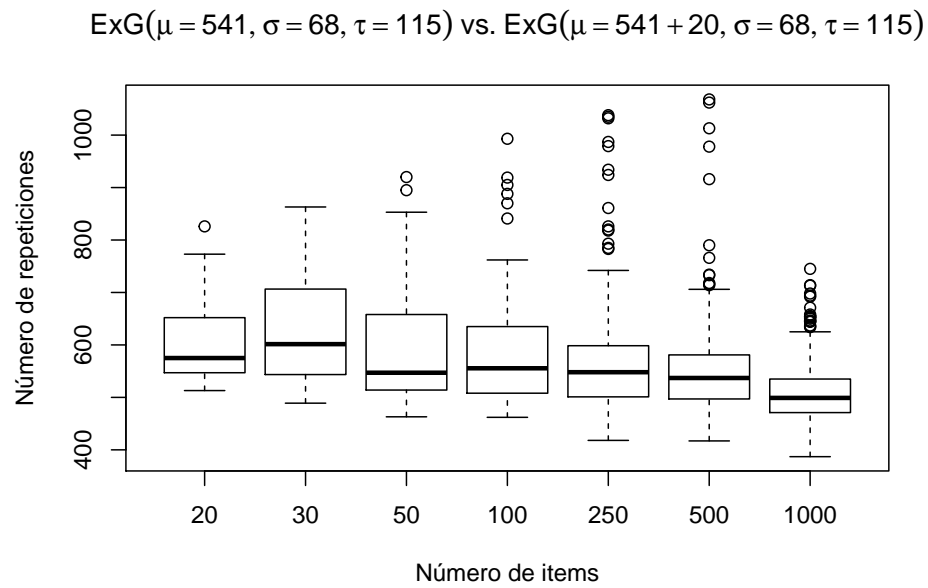


Figura 2.14: Mucha asimetría positiva. Efecto 20 ms. sobre la media.

$\text{ExG}(\mu = 541, \sigma = 68, \tau = 115)$  vs.  $\text{ExG}(\mu = 541 + 40, \sigma = 68, \tau = 115)$

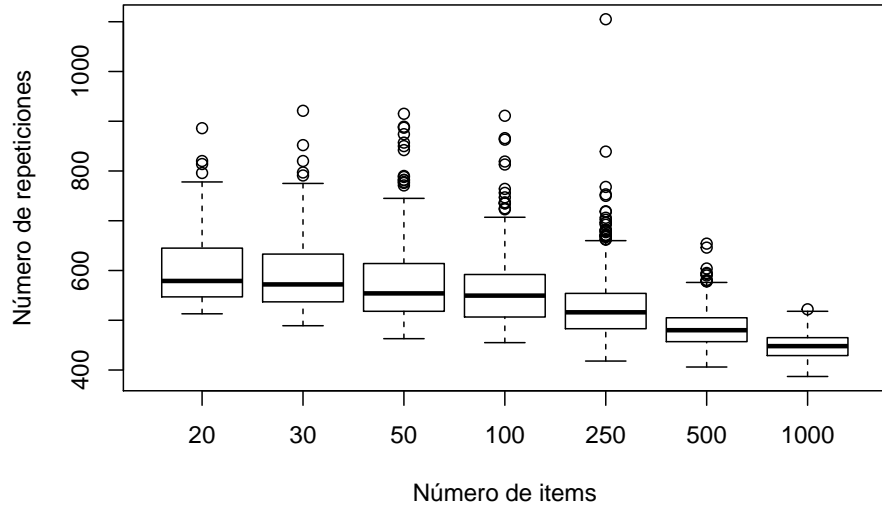


Figura 2.15: Mucha asimetría positiva. Efecto 40 ms. sobre la media.

$\text{ExG}(\mu = 541, \sigma = 68, \tau = 115)$  vs.  $\text{ExG}(\mu = 541 + 80, \sigma = 68, \tau = 115)$

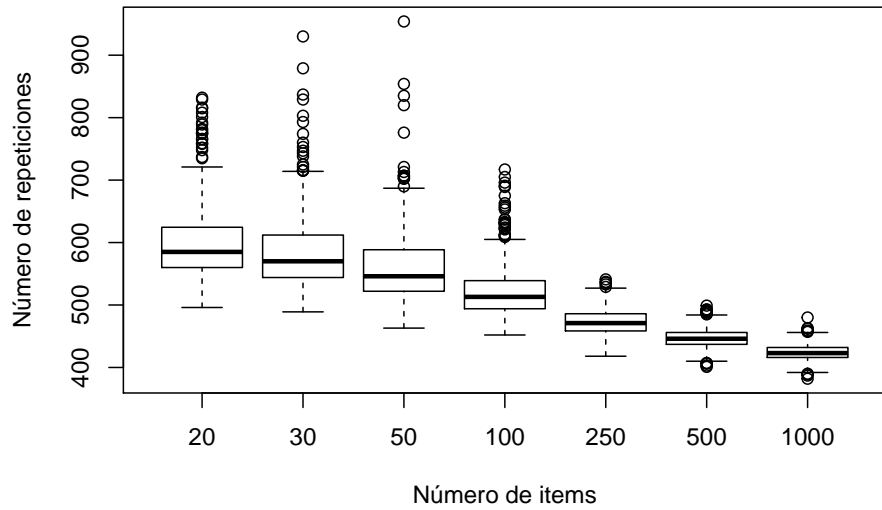


Figura 2.16: Mucha asimetría positiva. Efecto 80 ms. sobre la media.



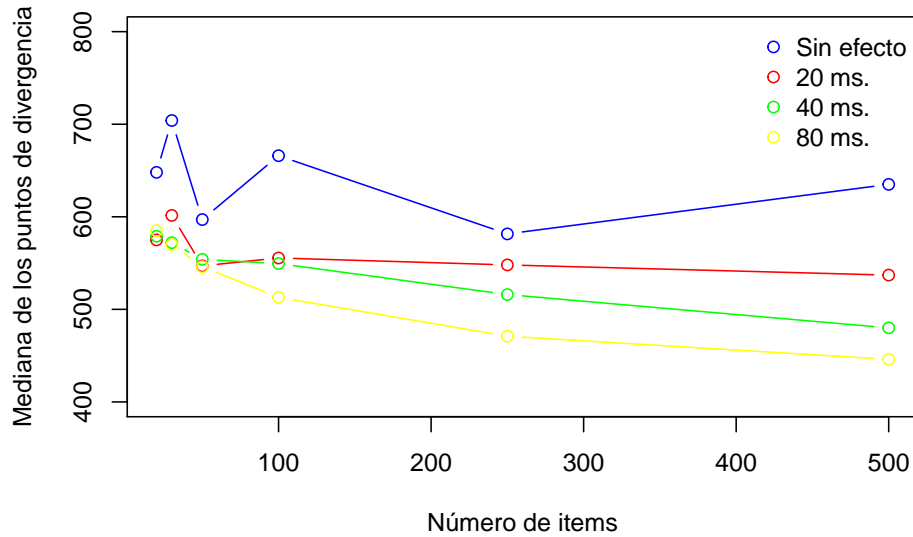


Figura 2.17: Análisis de potencia. Efectos sobre la media.

no se puede asegurar que un aumento en el número de ítems lleve aparejada una estabilidad en el punto de divergencia.

#### 2.3.2.4 Efecto sobre la cola

Con mucha asimetría positiva, aumentar el número de ítems parece incrementar el número de resultados fuera del diagrama de cajas. A la par, se reduce el valor de la media.

#### 2.3.2.5 Estudio de la mediana con efecto sobre la cola

No hay estabilidad ni uniformidad en la evolución del punto de divergencia al emplear efectos sobre la cola de la distribución.

## 2.4 Conclusiones

En una distribución de  $TR$  de tipo *ExGauss* la media de los  $TR$  disminuye al aumentar el efecto. Un aumento en el efecto, tanto sobre la media como sobre la

ExG( $\mu = 541, \sigma = 68, \tau = 115$ ) vs. ExG( $\mu = 541, \sigma = 68, \tau = 115 + 20$ )

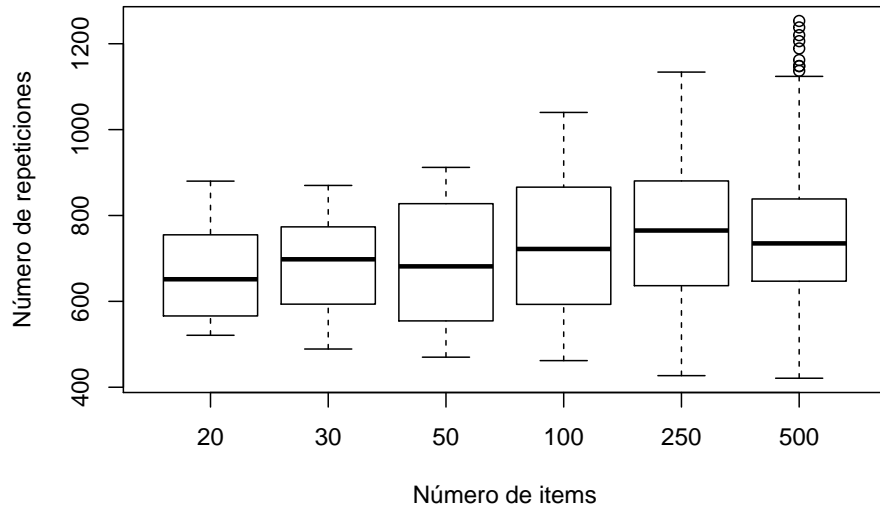


Figura 2.18: Mucha asimetría positiva. Efecto 20 ms. sobre la cola.

ExG( $\mu = 541, \sigma = 68, \tau = 115$ ) vs. ExG( $\mu = 541, \sigma = 68, \tau = 115 + 40$ )

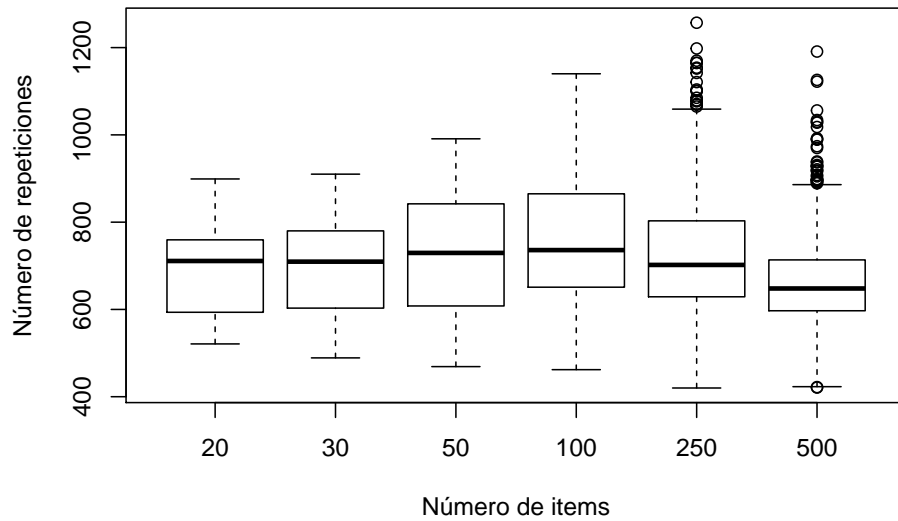


Figura 2.19: Mucha asimetría positiva. Efecto 40 ms. sobre la cola.

ExG( $\mu = 541, \sigma = 68, \tau = 115$ ) vs. ExG( $\mu = 541, \sigma = 68, \tau = 115 + 80$ )

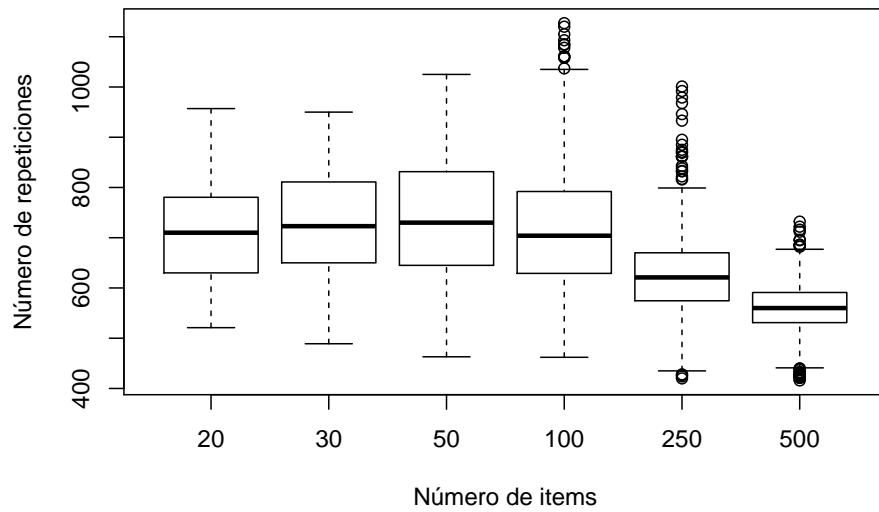


Figura 2.20: Mucha asimetría positiva. Efecto 80 ms. sobre la cola.

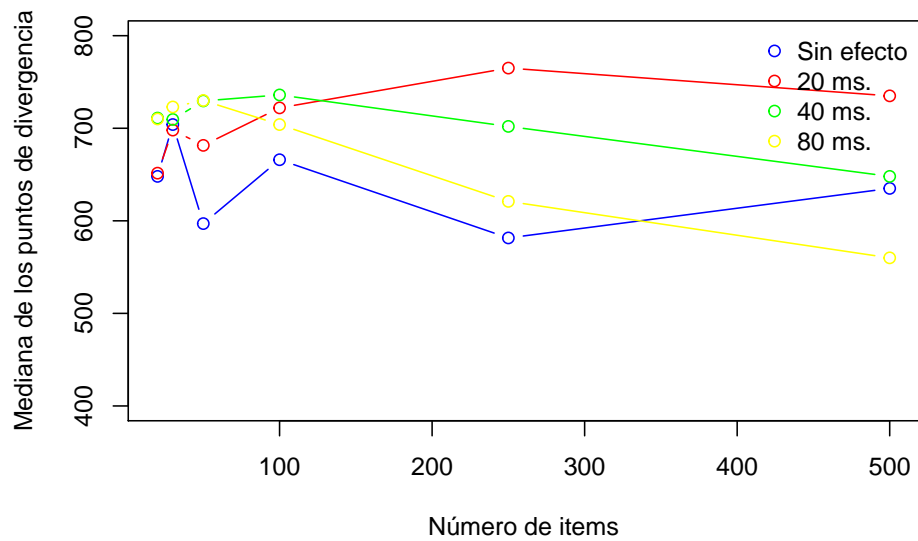


Figura 2.21: Análisis de potencia. Efectos sobre la cola.

cola de la distribución, aumenta el ritmo de descenso de dicha media. Además, el número de outliers se reduce al aumentar el efecto sobre la media pero parece aumentar al incrementarse el efecto sobre la cola.

El método propuesto por Reingold et al. (2012) no es estable para el cálculo del punto de divergencia y en las condiciones planteadas inicialmente por los autores no se trataría de un proceso robusto. Esto justificaría el trabajo posterior de Gomez et al. (2021) en que se explica formalmente las carencias teóricas del método. El trabajo de Gomez et al. (2021), en su edición online y como complemento al presente documento, puede leerse íntegro en la siguiente sección.

## Capítulo 3

# Are divergence point analyses suitable for response time data?<sup>1</sup>

### 3.1 Abstract

Estimating the time course of the influence of different factors in human performance is one of the major topics of research in cognitive psychology/neuroscience. Over the past decades, researchers have proposed several methods to tackle this question using latency data. Here we examine a recently proposed procedure that employs survival analyses on latency data to provide precise estimates of the timing of the first discernible influence of a given factor (e.g., word frequency on lexical access) on performance (e.g., fixation durations or response times). A number of articles have used this method in recent years, and hence an exploration of its strengths and its potential weaknesses is in order. Unfortunately, our analysis revealed that the technique has conceptual flaws, and it might lead researchers into believing that they are obtaining a measurement of processing components when, Latencies, Divergence, Mental

---

<sup>1</sup>Este capítulo ha sido publicado recientemente en la revista Behavior Research Methods (Decil 1 en JCR). La referencia es: Gomez, P., Breithaupt, J., Perea, M., & Rouder, J. N. (2021). Are divergence point analyses suitable for response time data? *Behavior Research Methods*, 53, 49–58. doi:10.3758/s13428-020-01424-1

chronometry in fact, they are obtaining an uninterpretable measurement.

### 3.2 Are divergence point analyses suitable for response time data?

Perhaps the most common cognitive psychology experiment is one in which participants are presented with stimuli that vary in a dimension of theoretical interest (e.g., stimuli repetition, word frequency, etc.). The stimulus elicits a response, and researchers measure response times (RTs) to draw inferences about hypothesized latent cognitive processes. This form of mental chronometry is commonly used in the analyses of data from a broad range of experimental paradigms such as choice tasks, naming, eye-tracking, and many others.

A critical intellectual endeavor is justifying the logic of the experimental inference—why specific data patterns inform conclusions about latent processes. The most popular model of analysis is tests of mean RTs, which are justified for fairly coarse conclusions. To draw a more fine-grained inference, however, researchers often consider the distributional properties of RTs (Balota & Yap, 2011; Heathcote et al., 1991; Ratcliff, 1979; Rouder et al., 2005). While more fine-grained conclusions are possible, inference based on distributional models often relies on secondary parametric assumptions (Luce, 1986) or commitments to specific processing models such as the diffusion model for choice response times (Ratcliff, 1978) or the EZ-reader model for eye fixation durations during reading (Reichle et al., 1998).

Recently, Sheridan and colleagues (Reingold, Reichle, Glaholt, & Sheridan, 2012; Reingold & Sheridan, 2014, 2018; Sheridan, 2013; Sheridan et al., 2013) proposed a novel mental chronometry method, termed divergence point analysis (DPA from now on), which is claimed to provide finer-grained descriptions of distributional effects without strong assumptions about functional form or a processing mechanism.

The setup for divergence point analysis applies to the comparison of two experimental conditions (e.g., low predictability vs. high predictability words). The dependent measure is a latency, and this latency can be for a manual response, an event-related component, or an eye-fixation duration. Divergence point analysis is applied to the distributions of latencies across the two conditions. In

the broadest definition, the point of divergence is the smallest response time value where the two distributions differ. Figure 1 provides an example in which the cumulative distribution functions (CDFs) are the same below the accentuated point but diverge after it<sup>2</sup>.

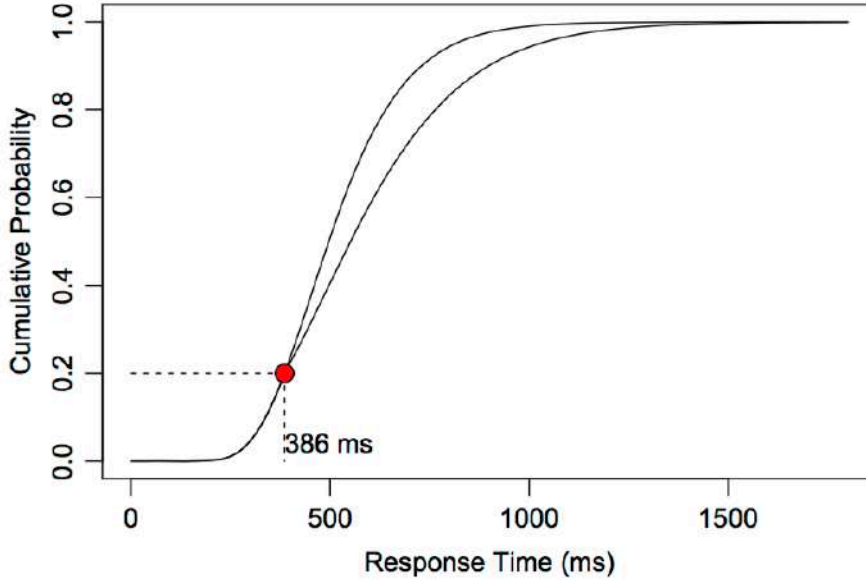


Figure 3.1: Example of a divergence point. Cumulative distribution functions (CDFs) of observed latencies for two conditions that are identical until the marked point. Thereafter, two functions diverge. This representation is typical of plots from Reingold and Sheridan (e.g., Figure 2 of Reingold & Sheridan, 2018)

Reingold et al. (2012) offered the following interpretation (in the context of their experiments on the effects of word frequency on eye fixation durations): “[The divergence point] might provide a promising and unique estimate for the earliest significant influence of word frequency on first fixation durations” (p. 185). By extension, the divergence point is the earliest point at which any manipulation first influences a response-time measure.

As we show below, our examination of divergence point analyses lead us to

---

<sup>2</sup>Reingold, Sheridan, and colleagues define divergence on survival functions rather than on cumulative distribution functions. The survival function  $S$  is  $1F$ , where  $F$  is the cumulative distribution function. Hence, divergence may be defined equivalently on survival or cumulative distribution functions (CDFs). We choose CDFs because we expect more readers are familiar with cumulative distribution functions than with survival functions.

conclude that they are difficult to use for inference about cognitive processes. There are two main problems in the DPA: (1) it has poor construct validity because in most cases the divergence point is trivial; and (2) it has poor statistical properties, as most empirically obtained divergence points are spurious.

### 3.3 Construct validity of divergence points

Reingold et al. (2012) refer to the divergence point as an estimate of first influence. Underlying all estimates is a concept of a true quantity. For example, when we compute the sample mean of a set of observations, we are estimating a true mean for the distribution underlying the sampling of these observations. Likewise, sample variances, sample percentiles, and sample effect sizes all have a true value in the underlying sampling process. Estimates serve as surrogates or best guesses at these true values, and these true values are the target of inference.

In some applications, however, it is difficult to know if the estimate corresponds to anything meaningful. Consider the concept of intelligence. It is always possible to administer tests and tally the scores. The act of tallying the scores, however, does not guarantee that the test measures intelligence or even that a true concept of intelligence exists (Gould, 1996). Whether a test measures its intended target is its construct validity, and construct validity stands as a critical part of measurement.

In the case of divergence point analyses, we believe that criteria for construct validity are in order. In our review of the method, we identified a basic criterion: There must be one true divergence point that the method estimates. Such a divergence point should be non-trivial, which means that it gives us information about the manipulation at hand. Figure 1 shows an example with good construct validity in that there is an actual divergence point that is not trivial. However, as will be shown below, such cases are rare, and not compatible with current information-processing theories.

#### 3.3.1 Trivial divergence points

The lack of construct validity is understood by considering the lower bound of support of a distribution. This lower bound refers to the point below which



latencies are not possible; for example, if latencies follow an exponential distribution that admits only positive response times, the lower bound is zero. Lower bounds need not be finite; for example, the lower bound for the normal and ex-Gaussian distributions are  $\infty$ . In most process models, the lower bound is a free parameter, called the irreducible minimum, which denotes a minimum time for encoding the stimulus and making a motor response (Dzhafarov, 1992; Luce, 1986; Ratcliff, 1979). Examples of such models include the diffusion model (Ratcliff, 1978) where the lower bound is  $T_{er}$ , race models (Rouder et al., 2005) where the lower bound is  $\psi$ , and the linear ballistic accumulator (Brown & Heathcote, 2008) where the lower bound is  $T_0$ .

We call a divergence point trivial if (a) both distributions share the same lower bound of support, and (b) the divergence point is at this lower bound of support. Consider the divergence point of two ex-Gaussian distributions that only differ in  $\tau$  (i.e., the tail parameter) while the values of  $\mu$  and  $\sigma$  are identical in both conditions. One might expect a divergence point in this setup, as the underlying factor seemingly affects a late component in the distribution. Figure 2a and b show the density functions and the CDFs, respectively, for this setup. Perhaps counter-intuitively, the CDFs have no common points to diverge from. Instead, the distribution with the smaller exponential scale is faster everywhere. If one is to talk about a divergence point at all, it would be at  $\infty$ , which in this case is the lower bound of support.

As it turns out, trivial divergence points are the norm in experimental situations. For example, this triviality holds in situations in which the  $\mu$  parameter is affected in the same direction as the  $\tau$  parameter by an experimental manipulation. This situation is quite ubiquitous, particularly for first-order effects like word frequency (see for example Table 1 in Staub, 2011, in which the frequency effects in first fixation duration are  $\mu = 16ms.$  , and  $\tau = 10ms.$ ).

Trivial divergence points are common in distributions with finite lower bounds of support. Take, for example, the diffusion model. Trivial divergence points occur when any parameter is affected, except for the  $T_{er}$ , the lower bound. The usual experiment is designed to affect drift rate, bound, or starting points, and these usual cases imply a trivial divergence.

Figure 3 shows some processes that result in trivial divergence. Each of the panels represents a process model with different assumptions regarding the source of variability in the latency distribution. In Fig. 3a, the first mechanism is the

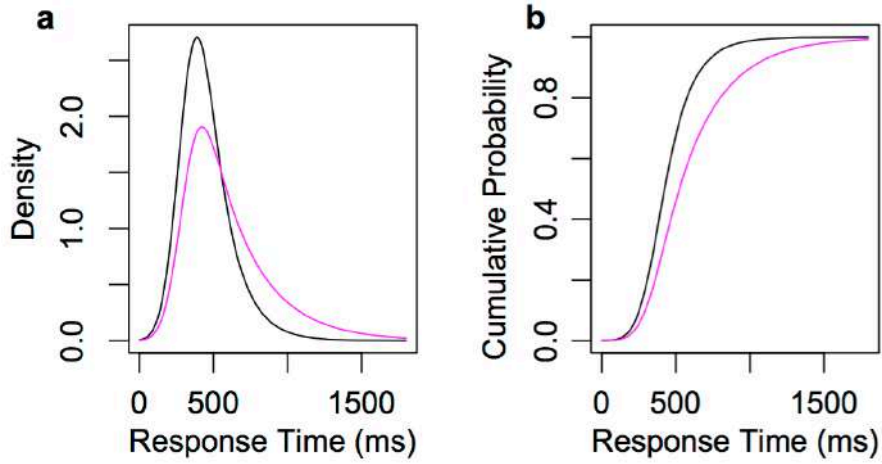


Figure 3.2: Density functions and cumulative distribution functions for ex-Gaussian distributions that have trivial divergence points. For both distributions  $\mu = 300$  and  $\sigma = 100$ ,  $\tau$  is 150 for the distribution is represented with *black lines*, and  $\tau = 300$  for the distribution represented with a *light line*

same for the two conditions, it has a mean of  $382\text{ ms.}$ , and a standard deviation of 50 (analogous to the  $T_0$  &  $S_T$  parameters in diffusion models). The beginning of the second process (the linear activation function) occurs at the end of the first process, and the latency is the point in time at which the activation reaches the threshold. Even if we assume that there is no variability in the activation rate within a category, the true divergence point is the shift value, which is trivial.

In the scenario represented by Fig. 3b, we changed the locus of variability. In this case, the ending of the first process is constant at  $382\text{ms.}$ , while the rate of activation of the second process varies within each condition. Again, the divergence point is trivial and corresponds to a cumulative probability value of zero. For Fig. 3c, we included variability in the ending time of the first process and variability in the activation level of the second process; again, while the two conditions are identical for the first component, and vary only on the second component, the divergence point is trivial and corresponds to a cumulative probability value of zero.

Figure 3d represents the assumption of activation beginning at time zero, and the second process affecting such activation only after the divergence point. We

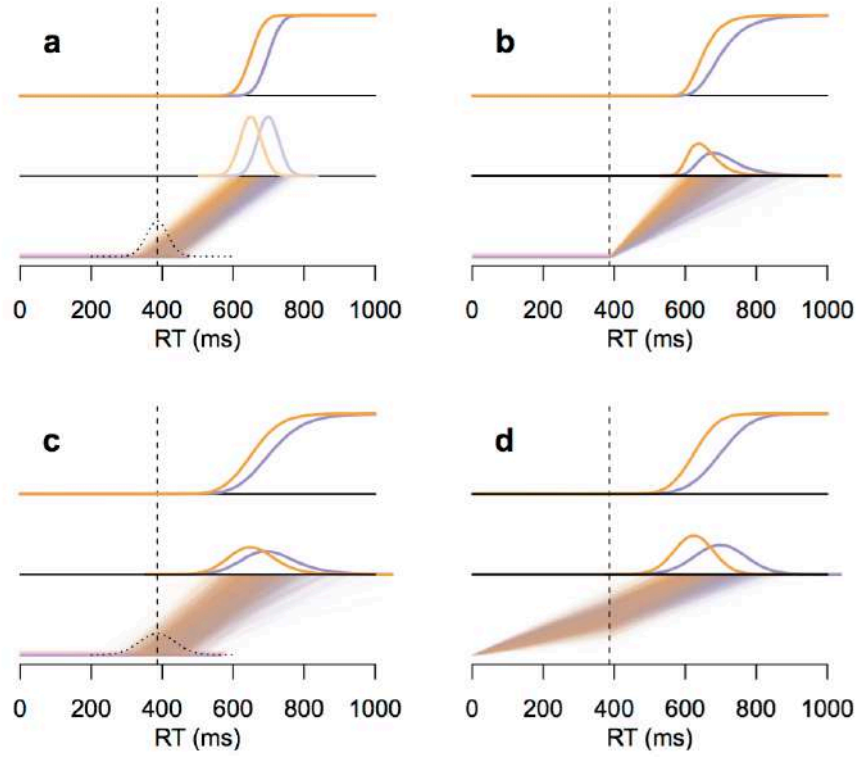


Figura 3.3: Four scenarios for plausible mechanisms of decision-making under a simplified linear activation function. For each of the panels, the *bottom part* represents the activation processes, the *middle part* represent the density function for the latencies, and the *top panel* represents the cumulative density functions. In the four panels, the distributions that have trivial divergence points. See the main text for an explanation of each of the mechanisms

assume some variability in the first process, but no difference between conditions, and a difference between conditions for the second process but without variability. In other words, the variability in this mechanism is restricted to the beginning activation. Again, the divergence point is trivial. Any combination of variability like the ones explored in Fig. 3a–d will generate a trivial divergence point. These examples raise the question of whether any process models yield a true non-trivial divergence points like that in Fig. 1.

### 3.3.2 Valid divergence points

The above cases show that typically there will only be trivial divergence point to analyze—that is, there is no construct validity to the estimate; however, we challenged ourselves to come up with a model where there was a true divergence point that was not trivial. We were able to generate cases such as those in Fig. 4a and b. Figure 4a is constructed as a horse race: Let the response latency be the fastest of two processes (represented by the dashed and the solid lines). One of the two processes is identically distributed in the two conditions (we will refer to it as the “D” process, as in the Dashed lines in the figure); the other process has, on average, lower activation rates and is represented in the figure by the solid lines (we will refer to it as the “S” process for the Solid lines in the figure). Critically, there is a difference between conditions for the “S” process. The shortest RTs come from the “D” process, and because this process does not vary between conditions, the density and the CDF are identical for the two conditions until a divergence point. This non-trivial divergence point emerges only under very specific parameter combinations and under the assumption of no other sources of variability in the latencies.

The second example of non-trivial divergence point is presented in Fig. 4b. In this case, the assumption is that the first process is identical between the two conditions and that the activation threshold can be reached with this process alone. The second process does not come into play until a later point in time (the vertical dashed line) and only if the threshold has not been reached. In this second process, there is a difference between the two conditions. As can be seen, this example does generate a non-trivial divergence point. Unfortunately, we believe that this is an unrealistic scenario. The density functions look unusual (see Fig. 4b). The reason is apparent—the divergence point marks a sudden change in derivative in the CDFs. This derivative of the CDF is the density

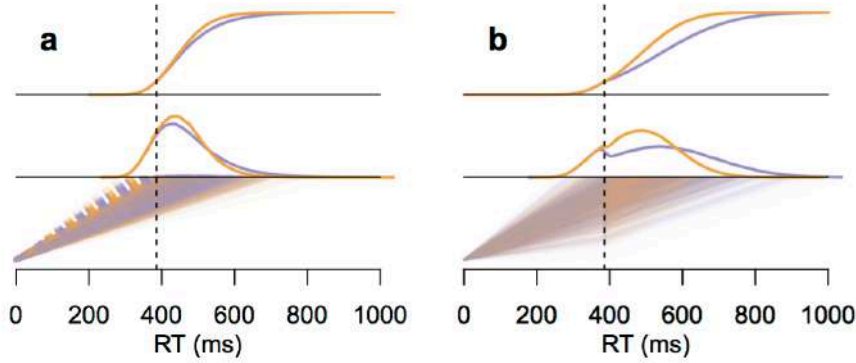


Figure 3.4: Density function and the cumulative distribution function for distributions that have a non-trivial divergence point. See main text for an explanation of each of the mechanisms

function, and these sudden changes imply violations of the absolute continuity of the density. The result is awkward distributions.

We can think, however, of some theoretical claims that could be consistent with Fig. 4b. In recognition memory tasks, some researchers have posited that familiarity is somewhat independent of recollection and that familiarity occurs earlier than recollection processes (Rotello & Heit, 1999). One could conceivably construct an experimental manipulation in which there are no familiarity differences between conditions, while there is a recollection advantage for one of the conditions. We suspect that in almost all other applications, the true divergence point will be trivial and that the estimate will lack construct validity. It remains an open challenge for proponents of the DPA to show that non-trivial true divergence points are plausible.

### 3.4 Statistical properties of divergence point analyses

In the previous section, we made a case for more in-depth scrutiny of the construct validity of DPA; in this section, we describe some problematic statistical properties that emerge from the lack of construct validity. First, we analyze the behavior of the DPA under the assumption of a null effect; then, we exami-

ne two scenarios that have been reported in the literature and appear to yield non-trivial divergence points.

### 3.4.1 Spurious divergence points

The lack of construct validity of DPA manifests itself as poor statistical properties that take slightly different forms depending on the specific implementation of the method. It is important to emphasize that we believe that these statistical issues are a consequence of the lack of construct validity in the method.

In Reingold & Sheridan’s (2014) bootstrapping method, in each iteration the latencies for each participant in each condition are randomly re-sampled with replacement. Each participant’s survival curves are then computed, to be averaged across subjects (a’ la Vincentile). Next, for each time bin  $t$ , the differences between conditions:  $t_i$  are computed ( $i$  stands for the number of iteration of the bootstrapping method), and then sorted. The range between the 5th and the 9995th value becomes the confidence interval  $CI(t)$  and the divergence point becomes the shortest  $t$  at which the  $CI(t)$  does not include 0. Researchers can decide the size of the bins.

A number of researchers have recently used this method (e.g., Leinenger 2018; Schmidtke & Kuperman 2019; Schmidtke, Matsuki, and Kuperman, 2017); however, Reingold & Sheridan (2014) abandoned this original bootstrapping method to find the first point of divergence. They realized that “. . . divergence point would be delayed relative to the actual point of divergence. This would be especially the case under low experimental power (i.e., a small number of participants and observations)” (p. 3), hence the bias of the bootstrapping method will not be discussed here.

The new method, termed ex-Gaussian DPA relies on a two-step process. In the first step, latencies are fit with ex- Gaussian distributions, and then the parameters are used to generate CDFs. When the CDFs vary by more than .015, the method detects a divergence point.

What happens if there is no true divergence because the true distributions for the two conditions are identical? We carried out a set of simulations with this scenario in mind. There are many ways to carry out these simulations with many parameters to vary; here, we present best-case scenarios with the

following assumptions: (1) for both conditions under comparison, the latencies were sampled from identical ex-Gaussian distributions; (2) although we varied the  $\mu$ ,  $\sigma$  and  $\tau$  parameters across simulations, within each simulation the values were identical for all participants; and (3) we varied the number of participants and number of items per condition.

Given that the latencies from both conditions come from the same distribution, there is no true divergence point (not even at  $T_0$ ). If the DPA produces a divergence point, we refer to it as a spurious divergence point. We carried out simulations with three levels of  $\mu$  values ( $50, 100, 150$ ), three levels of  $\sigma$  values ( $30, 60, 90$ ), three levels of  $\tau$  values ( $50, 100, 150$ ), four levels of number of trials per condition ( $20, 40, 80, 160$ ), and four levels of number of participants ( $20, 40, 80, 160$ ), for a total of 432 simulations ( $3 \times 3 \times 3 \times 4 \times 4 = 432$ ), for each combination of parameters, there were 500 simulated experiments. The RTs were generated with the `rexGAUS` function of the `gamlss.dist` package, and the ex-Gaussian fits were done with the `timefit` function of the `retimes` package in R.

Figure 5 shows the rate of spurious divergence points for different numbers of trials and participants and for the three levels of  $\mu$  and  $\tau$  (in the figure we present only the simulations with  $\sigma = 60ms$ ; other values of  $\sigma$  do not lead to different conclusions, but the complete set of simulations is available in the online appendix). In Table 1, we present an illustrative subset of the simulations ( $\mu = 100, \sigma = 60$  and  $\tau = 50$ ), the rate of spurious divergence points can be quite large, particularly for the number of trials and participants that are common and feasible in cognitive psychology experiments and psycholinguistics studies, in which there might be a limited number of possible materials to study the phenomenon of interest.

Cuadro 3.1: Rate of spurious divergence points for simulations with null effects. The latencies were generated with parameters  $\mu = 100$ ,  $\sigma = 60$ , and  $\tau = 50$  ms.

Number of participants	Number of trials	Rate of spurious DP	Mean DP
40	40	0.544	95
40	80	0.336	105
40	160	0.096	115
80	40	0.264	112

Number of participants	Number of trials	Rate of spurious DP	Mean DP
80	80	0.124	98
80	160	0.012	120
160	40	0.088	118
160	80	0.050	59
160	160	0.000	NA

The reason for such a large rate of spurious divergence points is straightforward: even small deviations in the sample data relative to the population parameters will yield differences in the ex-Gaussian parameters of the fits, and even very small differences in those parameters, for example a difference of a couple of milliseconds, will yield differences of .015 in the CDF/survival function. The ex-Gaussian fits are done on noisy data, and the method picks up the divergence in the noisy data. At large number of trials and participants, the DPA does a very good job of not returning a spurious point; the method does have consistency in that respect: as the number of observation increases, the method is less likely to give researchers an incorrect answer.

### 3.4.2 Divergence points that are CDF crossings

In recent work, Reingold & Sheridan (2018) have described CDF crossing points in simulation as divergence points. Thinking of these situations in terms of ex-Gaussian parameters is useful. Sheridan and Reingold consider cases with interacting effects where  $\mu$  and  $\tau$  are affected in opposite directions. Figure 6 is taken from the table in the Appendix of Sheridan and Reingold, and it shows a series of differences from their simulations. Each point is for a simulation run. Note how positive differences in one parameter are associated with negative differences in the other. The overlaid table shows, as an example, the values for one data point. In this chosen point, the value of the  $\mu$  parameter of the ex-Gaussian for the “slow” condition is smaller than the value for the “fast” condition. In contrast, the value of the  $\tau$  parameter is larger in the slow than in the fast condition. Such crossovers seem quite informative, as they indicate that the same manipulation has a facilitative effect in one component of processing, while it was an inhibitory effect in another.

The claim by Reingold and Sheridan is that these scenarios give rise to cases



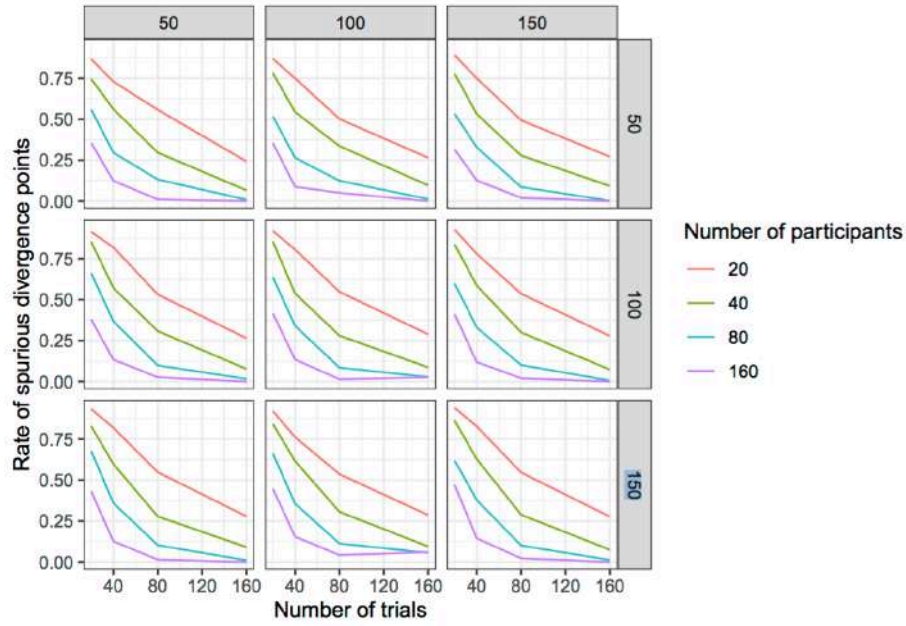


Figure 3.5: Different panels display the proportion of found divergence points assuming that there is no difference between the two conditions under comparison. The columns of panels represent the values of the mu parameter, and the rows represent the value of the tau parameter;  $\sigma = 60$  for all simulations. As would be expected, the rate of spurious divergence points is determined by the number of trials per condition (the x-axis within each panel), and the number of participants (the lines within each panel)

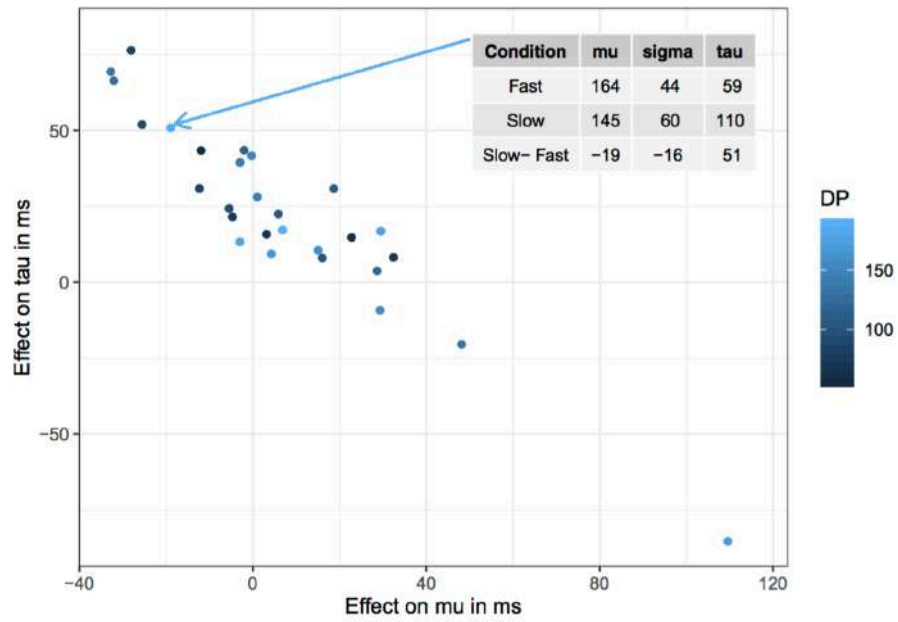


Figura 3.6: Effects on  $\mu$  and on  $\tau$  taken from Reingold and Sheridan (2018) Appendix 1. These were parameters used to find non-trivial divergence points in their simulations. While this would represent an interesting set of parameters in their own right, they are somewhat unusual; this is because across many experiments, the effects on  $\mu$  and on  $\tau$  tend to be positively correlated, not negatively correlated, as in the figure. The *shade of the points* relates to the location of the divergence point according to the DPA method

that have nontrivial divergence points, which are accurately localized by their divergence point analytic methods. We have three critiques of this. First, crossover points are not divergence points. They are not the earliest point at which distributions differ. The distributions differ throughout the range, and in this case, the divergence is trivial. Second, crossover points are exceedingly rare in practice. There are a few examples in the literature (Rouder, 2000; Yantis et al., 1991), but the vast majority of studies show no such crossover. Third, crossovers are diagnostic of a mixture of processes (Everitt & Hand, 1981; Falmagne, 1968). In such a scenario, the main theoretical implication is not the time of crossover, but what is the interpretation of the components and how are they affected by the manipulations at hand (See Fig. 7).

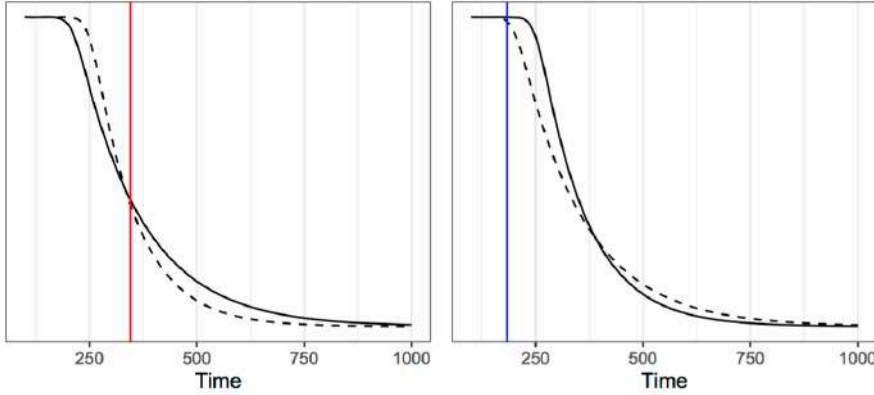


Figura 3.7: We present two examples of divergence points. Panels a and b show crossovers in the survival functions; the fast condition is shown with dashed lines

If researchers are looking for ways to describe these types of crossovers and dissociations between components of latency distributions, we believe that delta plots (e.g., De Jong, Liang, & Lauber, 1994; and particularly Ellinghaus & Miller, 2018 for delta plots with negative slopes) provide a far more cohesive and informative view.

Furthermore, the estimates from the DPA method in the crossover situation can be quite volatile. An example is provided by Reingold and Sheridan (2018) in their appendix for a divergence point of  $180ms$ . Figure 8 shows the survival functions using parameters similar to those in their table: for Condition 1:  $\mu_1 = 150$ ,  $\sigma_1 = 60$ ,  $\tau_1 = 78$ , and for Condition 2:  $\mu = 147$ ,  $\sigma_2 = 64$ ,  $\tau_2 = 91$ .

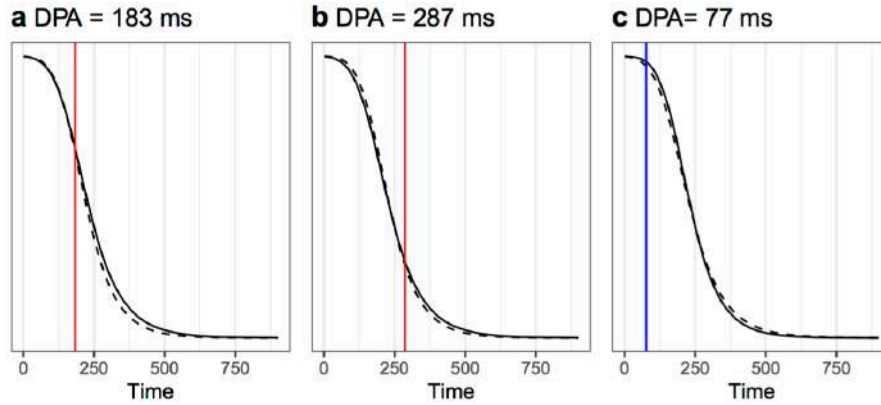


Figure 3.8: Illustration of how small changes can yield large fluctuations in the DPA results. The three panels show distributions with small variations in the  $\mu$  parameter ( $150ms.$  to  $160ms.$  to  $161ms.$ ) for one of the conditions, which yields widely different divergence points

Critically, even minuscule changes in the parameters generate radically different divergence points. For example, if the  $\mu$  parameter goes from  $150ms.$  to  $160ms.$ , to  $161ms.$ , the divergence point goes from  $183ms.$  to  $287ms.$  in favor of Condition 1, to  $75ms.$  in favor of Condition 2.

This finding complements the analysis shown in the previous subsection on spurious divergence points. Small variations due to sampling can produce radically different results, from spurious divergence points, to large swings in the value and direction of the divergence point.

### 3.5 Conclusions

Divergence points have limited interpretation. The apparent lack of true divergence points may seem counter-intuitive. For example, if we consider a reading task where participants decide if presented words are nouns or verbs, then we might expect that forming a representation in visual cortex occurs before the influence of the part-of-speech manipulation. This hypothetical might imply the plausibility of a divergence point. The part of speech manipulation does not affect the time course of processing before semantic meaning evaluation.

This hypothetical, which we find compelling, highlights the difficulty in interpretation. Divergence point analysis is not about moment-to-moment processing (see Estes, 1956 for another example of the difficulty in the interpretation of grouped data). It is possible, even likely, that there are manipulations that do not affect the moment-to-moment processing up to a specific point in time, after which there is a divergence in processing, yet, there are no divergence points in the collection of response time distributions. Divergence point analysis is about the response time distributions, and such a point cannot be interpreted in terms of latent moment-to-moment processing. Reingold and Sheridan (2018) acknowledge this issue; but this limitation is severe, and it constrains the usefulness and attractiveness of the method greatly. For the utmost transparency, users of the method should be explicit about these issues.

In most situations, the divergence point will be trivial, as in most experimental manipulations there is stochastic dominance: meaning that across all quantiles, the CDF for a fast condition will be to the left of the CDF for the slow condition regardless of what component (early vs. late) of processing is being tapped into. In the minority of experimental manipulations, there can be a dissociation between early components and late components; this can be easily implemented in process models like the EZ reader or evidence accumulation models. In these cases, the CDF might cross, and that crossing might be picked up by the DPA. We believe that if researchers want a purely descriptive, exploratory data analysis (EDA, Tukey, 1967) method, delta plots are better suited than DPA. Also, researchers could use linear mixed effects to explore the slope and intercept of the delta plots if inferential methods are in order (this, of course, deserves further examination).

In yet a smaller proportion of cases, there might be a true non-trivial divergence point. These situations are rare, and we had to think hard to come up with architectures that yield this type of effects. If the researcher’s model of the task is indeed one that generates a non-trivial divergence point, we believe that a DPA method might provide useful information. Note, however, that such architectures do not generate ex-Gaussian distributions, and hence fitting such functional form to the data might, or might not recover the true, non-trivial divergence point (we present examples in the online appendix). Nevertheless, we remain skeptical that such cases exist.

In sum, the divergence point analysis methods suffer from poor construct

validity, and hence, also of poor statistical properties. For these reasons, its usefulness is limited, and researchers should pause before using it.

## 3.6 Open practices statement

The code used to generate the figures and the simulations is available at <https://osf.io/ghw37/> R (Version 3.6.1, R Core Team, 2019).<sup>3</sup>

## 3.7 References

Aphalo, P.J. (2016). Learn R...as you learnt your mother tongue. Leanpub. Retrieved from <https://leanpub.com/learnr>.

Aust, F. (2018). papaja: Create APA manuscripts with R Markdown. Retrieved from <https://github.com/crsh/papaja>.

Balota, D.A., & Yap, M.J. (2011). Moving beyond the mean in studies of mental chronometry: The power of response time distributional analyses. *Current Directions in Psychological Science*, 20(3), 160–166.

Brown, S.D., & Heathcote, A. (2008). The simplest complete model of choice reaction time: Linear ballistic accumulation. *Cognitive Psychology*, 57, 153–178.

De Jong, R., Liang, C.C., & Lauber, E. (1994). Conditional and unconditional automaticity: A dual-process model of effects of spatial stimulus-response concordance. *Journal of Experimental Psychology: Human Perception and Performance*, 20, 731–750.

Dzhafarov, E.N. (1992). The structure of simple reaction time to step-function signals. *Journal of Mathematical Psychology*, 36, 235–268.

---

<sup>3</sup>We, furthermore, used the R-packages dplyr (Version 0.8.3, Wickham et al., 2019), forcats (Version 0.4.0, Wickham, 2019a, b), gamlss.dist (Version 5.1.4, Stasinopoulos & Rigby 2019), ggplot2 (Version 3.2.1, Wickham 2016), ggpmisc (Version 0.3.1, Aphalo, 2016), MASS (Version 7.3.51.4, Venables & Ripley 2002), papaja (Version 0.1.0.9842, Aust, (Aust, 2018)), purrr (Version 0.3.2, Henry & Wickham, 2019), readr (Version 1.3.1, Wickham et al., 2018), retimes (Version 0.1.2, Massidda, 2013), scales (Version 1.0.0, Wickham, 2018), stringr (Version 1.4.0, Wickham, 2019a, b), tibble (Version 2.1.3, Müller & Wickham 2019), tidyr (Version 0.8.3, Wickham & Henry, 2019), tidyverse (Version 1.2.1, Wickham, 2017), and truncnorm (Version 1.0.8; Mersmann, Trautmann, Steuer, and Bornkamp, 2018).

Ellinghaus, R., & Miller, J. (2018). Delta plots with negative- going slopes as a potential marker of decreasing response activation in masked semantic priming. *Psychological Research Psychologische Forschung*, 82(3), 590–599.

Estes, W.K. (1956). The problem of inference from curves based on group data. *Psychological Bulletin*, 53(2), 134–140.

Everitt, B.S., & Hand, D.J. (1981). Finite mixture distributions. London: Chapman; Hall. Falmagne, J.-C. (1968). Note on a simple fixed-point property of binary mixtures. *British Journal of Mathematical and Statistical Psychology*, 21, 131–132.

Gould, S.J. (1996). The mismeasure of man. New York: WW Norton & Company.

Heathcote, A., Popiel, S.J., & Mewhort, D. (1991). Analysis of response time distributions: An example using the Stroop task. *Psychological Bulletin*, 109(2), 340–347.

Henry, L., & Wickham, H. (2019). Purrr: Functional programming tools. Retrieved from <https://CRAN.R-project.org/package=purrr>.

Leinenger, M. (2018). Survival analyses reveal how early phonological processing affects eye movements during reading. *Journal of Experimental Psychology, Learning, Memory, and Cognition*.

Luce, R.D. (1986). Response times. New York: Oxford University Press.

Massidda, D. (2013). Retimes: Reaction time analysis. Retrieved from <https://CRAN.R-project.org/package=retimes>.

Mersmann, O., Trautmann, H., Steuer, D., & Bornkamp, B. (2018). Truncnorm: Truncated normal distribution. Retrieved from <https://CRAN.R-project.org/package=truncnorm>.

Müller, K., & Wickham, H. (2019). Tibble: Simple data frames .Retrieved from <https://CRAN.R-project.org/package=tibble>. Ratcliff, R. (1978). A theory of memory retrieval. *Psychological Review*, 85(2), 59–108.

Ratcliff, R. (1979). Group reaction time distributions and an analysis of distribution statistics. *Psychological Bulletin*, 86(3), 446–461. R Core Team (2019). R: A language and environment for statistical computing, Vienna, Austria. R

Foundation for Statistical Computing. Retrieved from <https://www.R-project.org/>.

Reichle, E.D., Pollatsek, A., Fisher, D.L., & Rayner, K. (1998). Toward a model of eye movement control in reading. *Psychological Review*, 105(1), 125–157.

Reingold, E.M., & Sheridan, H. (2014). Estimating the divergence point: A novel distributional analysis procedure for determining the onset of the influence of experimental variables. *Frontiers in Psychology*, 5, 1432.

Reingold, E.M., & Sheridan, H. (2018). On using distributional analysis techniques for determining the onset of the influence of experimental variables. *Quarterly Journal of Experimental Psychology*, 71(1), 260–271.

Reingold, E.M., Reichle, E.D., Glaholt, M.G., & Sheridan, H. (2012). Direct lexical control of eye movements in reading: Evidence from a survival analysis of fixation durations. *Cognitive Psychology*, 65(2), 177–206.

Rotello, C.M., & Heit, E. (1999). Two-process models of recognition memory: Evidence for recall-to-reject? *Journal of Memory and Language*, 40(3), 432–453.

Rouder, J.N. (2000). Assessing the roles of change discrimination and luminance integration: Evidence for a hybrid race model of perceptual decision making in luminance discrimination. *Journal of Experimental Psychology: Human Perception and Performance*, 26, 359–378.

Rouder, J.N., Lu, J., Speckman, P., Sun, D., & Jiang, Y. (2005). A hierarchical model for estimating response time distributions. *Psychonomic Bulletin & Review*, 12(2), 195–223.

Schmidtke, D., & Kuperman, V. (2019). A paradox of apparent brain- less behavior: The time-course of compound word recognition. *Cortex*, 116, 250–267.

Schmidtke, D., Matsuki, K., & Kuperman, V. (2017). Surviving blind decomposition: A distributional analysis of the time-course of complex word recognition. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 43(11), 1793.

Sheridan, H. (2013). The time-course of lexical influences on fixation durations during reading. Evidence from distributional analyses (PhD thesis).

Sheridan, H., Rayner, K., & Reingold, E.M. (2013). Unsegmented text de-



lays word identification: Evidence from a survival analysis of fixation durations. *Visual Cognition*, 21(1), 38–60.

Stasinopoulos, M., & Rigby, R. (2019). *Gamlss.dist: Distributions for generalized additive models for location scale and shape*. Retrieved from <https://CRAN.R-project.org/package=gamlss.dist>.

Staub, A. (2011). The effect of lexical predictability on distributions of eye fixation durations. *Psychonomic Bulletin & Review*, 18(2), 371–376.

Venables, W.N., & Ripley, B.D. (2002). *Modern applied statistics with S* (Fourth.), Springer, New York. Retrieved from <http://www.stats.ox.ac.uk/pub/MASS4>.

Wickham, H. (2016). *Ggplot2: Elegant graphics for data analysis*. New York: Springer. Retrieved from <https://ggplot2.tidyverse.org>.

Wickham, H. (2017). *Tidyverse: Easily install and load the ‘tidyverse’*. Retrieved from <https://CRAN.R-project.org/package=tidyverse>.

Wickham, H. (2018). *Scales: Scale functions for visualization*. Retrieved from <https://CRAN.R-project.org/package=scales>.

Wickham, H. (2019a). *Forcats: Tools for working with categorical variables (factors)*. Retrieved from <https://CRAN.R-project.org/package=forcats>.

Wickham, H. (2019b). *Stringr: Simple, consistent wrappers for common string operations*. Retrieved from <https://CRAN.R-project.org/package=stringr>.

Wickham, H., & Henry, L. (2019). *Tidyr: Easily tidy data with ‘spread()’ and ‘gather()’ functions*. Retrieved from <https://CRAN.R-project.org/package=tidyr>.

Wickham, H., Hester, J., & Francois, R. (2018). *Readr: Read rectangular text data*. Retrieved from <https://CRAN.R-project.org/package=readr>.

Wickham, H., Francois, R., Henry, L., Müller, K. (2019). *Dplyr: A grammar of data manipulation*. Retrieved from <https://CRAN.R-project.org/package=dplyr>.

Yantis, S., Meyer, D.E., & Smith, J.E.K. (1991). Analysis of multinomial mixture distributions: New tests for stochastic models of cognitive action. *Psychological Bulletin*, 110, 350–374.



## Capítulo 4

# Redes neuronales aplicadas al reconocimiento de palabras. Un modelo autoorganizado de aprendizaje lector

### 4.1 Fundamentación del modelo

En Kohonen (1990) se propuso un modelo genérico de red neuronal denominado *Mapa Autoorganizado* o *Self Organizing Map*, que abreviaremos *SOM*. Se trata de un modelo de aprendizaje competitivo no organizado.

El modelo planteado trata de emular los procesos biológicos según los cuáles las neuronas, sometidas a diferentes estímulos, reaccionan en mayor o menor medida a los mismos en un proceso de aprendizaje continuo.

En el trabajo Polk, Lacey, Nelson, Demiralp, Newman, Kraus, Raheja & Farah (2009) se planteó el *SOM* como una opción para el estudio de aprendizaje lector. Pongamos por ejemplo las representaciones gráficas de “a” y “A”. El

aspecto visual de ambas grafías es totalmente diferente pero el concepto debe llegar a ser el mismo al final del proceso de aprendizaje. El sentido común apunta a que por otra parte, las representaciones “o” y “O” alcanzarán rápidamente el mismo concepto.

Los grados de similitud entre los pares minúscula / mayúscula se puede estudiar con mayor detalle en Boles & Clifford (1989) o Lerner, Amstrong & Frost (2014).

En términos generales:

- Representaciones gráficas similares deberían activar patrones neuronales similares.
- Las parejas minúscula / mayúscula de la misma letra (*a/A, b/B, c/C*, etc) deben alcanzar el mismo concepto o *Abstract Letter Identity*, en adelante *ALI*.

El cuerpo del trabajo que sigue se organizará en tres partes diferenciadas:

- Entrenamiento de mapa de letras.
- Entrenamiento de mapa de palabras.
- Análisis del error.

## 4.2 Datos

Los entrenamientos, tanto de letras como de palabras, toman como información de entrada unos vectores pesos para las diferentes letras (una palabra es en esencia una concatenación de letras). La forma de determinar estos vectores pesos de letras plantea diferentes grados de bondad del análisis siguiente.

Cuadro 4.1: Letras. Estímulos de entrenamiento.

a	b	c	d	e	f	g	h	i	j	k	l	m
n	o	p	q	r	s	t	u	v	w	x	y	z
A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Cuadro 4.2: Palabras de tres letras. Estímulos de entrenamiento.

bdz	BDZ	BKZ	zao	ZAO	ZKO	vao	VAO	VJO
xqi	XQI	XTI	orw	ORW	OKW	sqv	SQV	SKV
zgv	ZGV	ZHV	vrw	VRW	VKW	wds	WDS	WKS
bgp	BGP	BTP	oaz	OAZ	OTZ	pdo	PDO	PHO
pev	PEV	PKV	sri	SRI	SKI	pex	PEX	PJX
wdv	WDV	WJV	vrb	VRB	VTB	iev	IEV	IKV
pqi	PQI	PTI	sez	SEZ	SKZ	bdv	BDV	BKV
idp	IDP	IKP	wqz	WQZ	WKZ	wdi	WDI	WKI
ieo	IEO	IHO	vgz	VGZ	VHZ	irs	IRS	IKS
sqw	SQW	STW	irw	IRW	ITW	bqx	BQX	BKX
xao	XAO	XKO	sep	SEP	STP	pdz	PDZ	PKZ
pgw	PGW	PKW	bgw	BGW	BJW	oax	OAX	OKX
zqi	ZQI	ZJI	oav	OAV	OKV	idb	IDB	IHB
vri	VRI	VTI	sax	SAX	SKX	xrv	XRv	XKV
xeb	XEB	XKB	zax	ZAX	ZKX	bgs	BGS	BKS
ogp	OGP	OTP	wqx	WQX	WTX	xaz	XAZ	XTZ
bqi	BQI	BTI	xas	XAS	XTS	zab	ZAB	ZKB
oei	OEI	OKI	vax	VAX	VHX	srb	SRB	STB
web	WEB	WTB	wqp	WQP	WKP	zep	ZEP	ZTP
pgb	PGB	PJB	zgs	ZGS	ZTS	vdv	VDP	VTP
ogs	OGS	OKS	weo	WEO	WTO	igz	IGZ	IKZ
beo	BEO	BJO	vqs	VQS	VTs	xrw	XRW	XHW
odb	ODB	OTB	xgp	XGP	XJP	sro	SRO	STO
idx	IDX	ITX	prs	PRS	PKS	zqw	ZQW	ZHW

Nótese que el listado de palabras de entrenamiento del modelo se ha extraído de Polk et al. (2009) Este será el trabajo de partida del desarrollo siguiente. Otros trabajos del mismo autor en que se desarrolla la misma base y se presentan los resultados del mapa entrenado son Polk & Farah (1997) y Polk, Stallcupm, Aguirre, Alsop, D’Esposito, Dentre & Farah (2002).

Para la generación de pesos de las letras planteamos tres posibles opciones:

- **Método 1:** Vectores aleatorios 5-dimensionales en  $Un(0,1)$ .
- **Método 2:** Lectura visual de las grafías. Lectura de imágenes  $15 \times 15$  de

las 52 representaciones visuales de letras. Se convierte cada pixel es una componente de un vector 225-dimensional. El valor de cada componente es un color en formato hexadecimal. Si dicho color es blanco #FFFFFF se asigna el valor 0 a la misma componente y en caso contrario 1.

- **Método 3:** Incorporación del componente fonético. Los vectores 225-dimensionales generados en el Método 2 se amplían hasta 250 incorporando 25 componentes (a penas un 10 % más) de forma que las representaciones minúscula y mayúscula emparejadas, aun pudiendo tener representaciones visuales claramente diferenciadas, presenten estas últimas 25 componentes aleatorias idénticas (*a* y *A* presentan clara diferenciación visual pero se corresponden con el mismo sonido). Esta parte aporta una novedad sobre la base de Polk et al. (2009).

```
som1generador<-function(metodogen='r')
{
  # Defino el alfabeto y los códigos
  abecedario<-as.vector(read.table("letras.txt")$V1);
  nletras<-length(abecedario);

  if(metodogen=='r') # Generación random, peso tamaño 5 por defecto
  {
    tam=5;
    abecedario_pesos<-matrix(data=runif(n=nletras*tam,min=0,max=1),
      nrow=nletras,ncol=tam);
  }

  if(metodogen=='v') # Lectura de imagenes (view)
  {
    # Las letras/est?mulos son vectores de 225 componentes
    # provenientes de leer 56 im?genes de las diferentes
    # letras de bmp y trasladarlo a vector.
    # El procedimiento no está pormenorizado en (Polk, 2009).

    letras_code2<-matrix(rep(0,nletras*225),nrow=nletras,ncol=225);
    for(i in 1:nletras)
    {
      if(i<=nletras)
```

```

{
  letras_code2[i,]<-read.bmp(f=paste("resized_",abecedario[i],
".bmp",sep=""));
  letras_code2[i,]<-as.vector(as.raster(letras_code2[i,],
max=255));
}
if(i>nletras)
{
  letras_code2[i,]<-read.bmp(f=paste("resized_",abecedario[i],
'2'," .bmp",sep=""));
  letras_code2[2,]<-as.vector(as.raster(letras_code2[i,],
max=255));
}
}
abecedario_pesos<-matrix(rep(0,nletras*225),nrow=nletras,ncol=225);
for(i in 1:nletras)
{
  for(j in 1:225)
  {
    if(letras_code2[i,j]=="#FFFFFF")
    {
      abecedario_pesos[i,j]<-0;
    }
    else
    {
      abecedario_pesos[i,j]<-1;
    }
  }
}
}

# Lectura de imagenes (view) y fonética (f)
if(metodogen=="vf")
{
  abecedario_pesos<-matrix(nrow=nletras,ncol=250);

```

```

for(i in seq(from=1,to=0.5*nletras,by=1))
{
  fono<-runif(n=25,min=0,max=1);

  abecedario_pesos[i,<-read.bmp(f=paste("resized_",abecedario[i],
".bmp",sep=""));
  abecedario_pesos[i,<-as.vector(as.raster(abecedario_pesos[i,],
max=255));

  abecedario_pesos[i,<-c(abecedario_pesos[i,],fono);

  abecedario_pesos[i+0.5*nletras,<-c(as.vector(
as.raster(read.bmp(f=paste("resized_",abecedario[i],'2'," .bmp",
sep="")),max=255)),fono);
}
# subíndice fuera de los límites

for(i in 1:nletras)
{
  for(j in 1:225)
  {
    if(abecedario_pesos[i,j]=="#FFFFFF")
    {
      abecedario_pesos[i,j]<-0;
    }
    else
    {
      abecedario_pesos[i,j]<-1;
    }
  }
}
}

write(x=abecedario_pesos,file="abecedario.txt");
}

```



## 4.3 Algoritmos y código fuente

### 4.3.1 Algoritmo original

Extracto del trabajo de Polk et al.(2009)

*SOM learning is accomplished by modifying the weight vector of the winning cell so that it is more similar to the input, therefore making this cell more likely to win again with future presentations. Critically, the weight vectors of cells in close spatial proximity to the winning cells are also updated. As a result, with experience the response of a SOM tends to become spatially organized, learning a mapping from the statistical regularities discovered in its k-dimensional input space to a set of topographically organized neighbourhoods within the SOM. Mathematically, if  $x(t)$  is a vector representing the input to a SOM at time  $t$ , and  $w_i(t)$  represents the weight vector of cell  $i$  at time  $t$ , then the winning cell  $c$  is given by*

$$\text{ArgMin}_c \|x(t) - w_c(t)\|$$

*where the  $\|$  operator denotes vector distance. We use Euclidean distance for all distance computations in the model. The weight vector for each cell  $i$  is updated according to the following SOM learning equation:*

$$w_i(t+1) = w_i(t) + \alpha(t)h_{c,i}(t)(x(t) - w_c(t))$$

*where  $\alpha(t)$  is a time-dependent learning rate, and  $h_{c,i}(t)$  is a kernel function that is centred on the winning cell  $c$  and that computes the magnitude of the update to cell  $i$  based on its spatial proximity to the winning cell. In our model we use the Gaussian kernel function*

$$h_{c,i}(t) = e^{-\frac{1}{\lambda(t)^2} \|i(x,y) - c(x,y)\|^2}$$

*,where  $\lambda(t)$  is a time-dependent parameter that determines the width of the kernel, and  $i(x,y)$  and  $c(x,y)$  denote the map coordinates of cells  $i$  and  $c$ , respectively. We compute  $\alpha(t)$  and  $\lambda(t)$  as exponentially decreasing functions of time, with time denoting discrete presentations of training inputs.*

Extracto de Kohonen (1990)

*The basic idea underlying what is called competitive learning is roughly as*

follows: Assume a sequence of statistical samples of a vectorial observable  $x = x(t) \in \mathbb{R}^n$ , where  $t$  is the time coordinate, and a set of variable reference vectors  $\{m_i(t) : m_i \in \mathbb{R}^n, i = 1, 2, \dots, k\}$ . Assume that the  $m_i(0)$  have been initialized in some proper way; random selection will often suffice. If  $x(t)$  can somehow be simultaneously compared with each  $m_i(t)$  at each successive instant of time, taken here to be an integer  $t = 1, 2, 3, \dots$ , then the best-matching  $m_i(t)$  is to be updated to match even more closely the current  $x(t)$ . If the comparison is based on some distance measure  $d(x, m_i)$  altering  $m_i$  must be such that, if  $i = c$  is the index of the best-matching reference vector, then  $d(x, m_c)$  is decreased, and all the other reference vectors  $m_i$ , with  $i \neq c$ , are left intact. In this way the different reference vectors tend to become specifically “tuned” to different domains of the input variable  $x$ , then the  $m_i$  tend to be located in the input space  $\mathbb{R}^n$  in such a way that they approximate to  $p$  in the sense of some minimal residual error.

Vector Quantization (VQ) is a classical method, that produces an approximation to a continuous probability density function  $p(x)$  of the vectorial input variable  $x$  using a finite number of codebook vectors  $m_i, i = 1, 2, \dots, k$ . Once the “codebook” is chosen, the approximation of  $x$  involves finding the reference vector  $m_c$  closest to  $x$ . One kind of optimal placement of the  $m_i$  minimizes  $E$ , the expected  $r$ th power of the reconstruction error:

$$E = \int \|x - m_c\|^r p(x) dx$$

where  $dx$  is the volume differential in the  $x$  space, and the index  $c = c(x)$  of the best-matching codebook vector (“winner”) is a function of the input vector  $x$ :

$$\|x - m_c\| = \min_i \|x - m_i\|$$

In general, no closed-form solution for the optimal placement of the  $m_i$  is possible, and iterative approximation schemes must be used.

(...)

Using the square-error criterion  $r = 2$ , It can also be shown that the following stepwise “delta rule”, in the discrete-time formalism  $t = 0, 1, 2, \dots$ , defines the optimal values asymptotically. Let  $m_c = m_c(t)$  be the closest codebook vector to  $x = x(t)$  in the Euclidian metric. The steepest-descent gradient-step optimiza-

tion of  $E$  in the  $m_c$  space yields the sequence

$$\begin{cases} m_c(t+1) = m_c(t) + \alpha(t)[x(t) - m_c(t)] \\ m_i(t+1) = m_i(t) \end{cases} \quad \text{for } i \neq c$$

with  $\alpha(t)$  being a suitable, monotonically decreasing sequence of scalar-valued gain coefficients,  $0 < \alpha(t) < 1$ . This is then the simplest analytical description of competitive learning.

In general, if we express the dissimilarity of  $x$  and  $m_i$  in terms of a general distance function  $d(x, m_i)$ , we have first to identify the “winner”  $m_c$  such that

$$d(x, m_c) = \min_i d(x, m_i)$$

. After that updating rule should be used such that decreases monitically: the correction  $\delta(m_i)$  of  $m_i$  must be such that

$$[\text{grad}_{m_i} d(x, m_i)]^T \delta(m_i) < 0$$

.  
El algoritmo base empleado es el de Polk et al.(2009) pero se ha tenido en cuenta la puntualización de Kohonen (1990) referida a no actualizar el peso de la neurona activa / ganadora frente al estímulo presentado en cada iteración.

### 4.3.2 Códigos fuente

```
som1<-function(filas_cerebro=10,columnas_cerebro=10,altura_cerebro=5,nrep=1000)
{
  # Defino el alfabeto y los códigos
  abecedario<-c('a','b','c','d','e','f','g','h','i','j','k','l','m',
    'n','o','p','q','r','s','t','u','v','w','x','y','z','A','B','C','D',
    'E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U',
    'V','W','X','Y','Z');
  nletras<-length(abecedario);
  som1generador(tam=altura_cerebro);
  abecedario_pesos<-matrix(scan(file="abecedario.txt"),nrow=nletras,
    ncol=altura_cerebro,byrow=TRUE);
```

```

cerebro<-array(data=runif(n=filas_cerebro*columnas_cerebro*altura_cerebro,
min=0,max=1),
dim=c(filas_cerebro,columnas_cerebro,altura_cerebro));

r0<-0.5*max(filas_cerebro,columnas_cerebro);
lambda<-nrep/log(r0);
aprendizaje<-0.9;

for(t in 1:nrep)
{
  i0<-1;
  j0<-1;
  neurona_activa<-cerebro[i0,j0,];
  estimulo<-abecedario_pesos[sample(x=1:nletras,size=1),];
  for(i in 1:filas_cerebro)
  {
    for(j in 1:columnas_cerebro)
    {
      celula<-cerebro[i,j,];
      if(norm(x=celula-estimulo,type="2")<norm(x=celula-neurona_activa,
type="2"))
      {
        i0<-i;
        j0<-j;
        neurona_activa<-celula;
      }
    }
  }

  r<-r0*exp(-t/lambda);
  aprendizaje<-aprendizaje*exp(-t/lambda);

  for(i in 1:filas_cerebro)
  {
    for(j in 1:columnas_cerebro)
    {

```

```

        if(i!=i0 & j!=j0)
        {
            # Distancia a la célula activa
            d<-dist(matrix(c(i0,i,j0,j),2,2));
            entorno<-exp(-(d^2)/(2*(r^2)));
            gamma<-entorno*aprendizaje;

            cerebro[i,j,<-cerebro[i,j,]+gamma*(estimulo-cerebro[i0,j0,]);
        }
    }
}

return(cerebro);
}

```

```

som1g<-function(filas_cerebro=10,columnas_cerebro=10,altura_cerebro=5,
nrep=1000,cerebro,letra='a')
{
    abecedario<-c('a','b','c','d','e','f','g','h','i','j','k','l','m',
        'n','o','p','q','r','s','t','u','v','w','x','y','z','A','B','C','D',
        'E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U',
        'V','W','X','Y','Z');
    nletras<-length(abecedario);

    cerebro<-som1(filas_cerebro=10,columnas_cerebro=10,
        altura_cerebro=5,nrep=1000);

    abecedario_pesos<-matrix(scan(file="abecedario.txt"),
        nrow=nletras,ncol=altura_cerebro,byrow=TRUE);

    letra_peso<-abecedario_pesos[which(abecedario==letra),];
    matriz_activacion<-matrix(nrow=filas_cerebro,ncol=columnas_cerebro);

    aux<-matrix(nrow=2,ncol=altura_cerebro);
    aux[1,<-letra_peso;
    for(i in 1:filas_cerebro)

```

```

{
  for(j in 1:columnas_cerebro)
  {
    aux[2,]<-cerebro[i,j,];
    matriz_activacion[i,j]<-dist(x=aux);
  }
}
lista_activacion<-sort(as.vector(matriz_activacion));

matriz_colores<-matrix(data=rep(0,filas_cerebro*columnas_cerebro),
nrow=filas_cerebro,ncol=columnas_cerebro);

for(i in 1:filas_cerebro)
{
  for(j in 1:columnas_cerebro)
  {
    if(matriz_activacion[i,j]>=quantile(lista_activacion,0.75))
    {
      matriz_colores[i,j]<-"gray100";
    }
    if(matriz_activacion[i,j]>median(lista_activacion) & matriz_activacion[i,j]<quantile(lista_activacion,0.75))
    {
      matriz_colores[i,j]<-"gray70";
    }
    if(matriz_activacion[i,j]>quantile(lista_activacion,0.25) & matriz_activacion[i,j]<median(lista_activacion))
    {
      matriz_colores[i,j]<-"gray40";
    }
    if(matriz_activacion[i,j]<=quantile(lista_activacion,0.25))
    {
      matriz_colores[i,j]<-"gray0";
    }
  }
}

barplot(rep(10,columnas_cerebro),border=NA,axes=FALSE,space=0,

```

```

col=matriz_colores[1,],ylim=c(1,filas_cerebro),main=letra);

for(i in seq(from=filas_cerebro-1,to=1,by=-1))
{
  barplot(rep(i,columnas_cerebro),border=NA,axes=FALSE,space=0,
    col=matriz_colores[i,],add=TRUE);
}
}

som1hist<-function(cerebro,letra='a')
{
  abecedario<-c('a','b','c','d','e','f','g','h','i','j','k','l','m',
    'n','o','p','q','r','s','t','u','v','w','x','y','z','A','B','C','D',
    'E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U',
    'V','W','X','Y','Z');
  nletras<-length(abecedario);

  filas_cerebro<-dim(cerebro)[1];
  columnas_cerebro<-dim(cerebro)[2];
  altura_cerebro<-dim(cerebro)[3];

  abecedario_pesos<-matrix(scan(file="abecedario.txt"),nrow=nletras,
    ncol=altura_cerebro,byrow=TRUE);

  letra_peso<-abecedario_pesos[which(abecedario==letra),];
  matriz_activacion<-matrix(nrow=filas_cerebro,ncol=columnas_cerebro);

  aux<-matrix(nrow=2,ncol=altura_cerebro);
  aux[1,<-letra_peso;
  for(i in 1:filas_cerebro)
  {
    for(j in 1:columnas_cerebro)
    {
      aux[2,<-cerebro[i,j,];
      matriz_activacion[i,j]<-dist(x=aux);
    }
  }
}

```

```

lista_activacion<-sort(as.vector(matriz_activacion));

matriz_colores<-matrix(data=rep(0,filas_cerebro*columnas_cerebro),
nrow=filas_cerebro,ncol=columnas_cerebro);

izq<-min(lista_activacion);
rango<-max(lista_activacion)-min(lista_activacion);
for(i in 1:filas_cerebro)
{
  for(j in 1:columnas_cerebro)
  {
    if(matriz_activacion[i,j]>=izq+0.75*rango)
    {
      matriz_colores[i,j]<-"gray100";
    }
    if(matriz_activacion[i,j]>izq+0.5*rango &
matriz_activacion[i,j]<izq+0.75*rango)
    {
      matriz_colores[i,j]<-"gray70";
    }
    if(matriz_activacion[i,j]>izq+0.25*rango &
matriz_activacion[i,j]<=izq+0.5*rango)
    {
      matriz_colores[i,j]<-"gray40";
    }
    if(matriz_activacion[i,j]<=izq+0.25*rango)
    {
      matriz_colores[i,j]<-"gray0";
    }
  }
}

hist3D(x=seq(from=0,to=filas_cerebro-1,by=1),
y=seq(from=0,to=columnas_cerebro-1,by=1),
z=matriz_activacion,col=matriz_colores);
}

```



```

som1g<-function(cerebro,letra='a')
{
  abecedario<-c('a','b','c','d','e','f','g','h','i','j','k','l','m',
    'n','o','p','q','r','s','t','u','v','w','x','y','z','A','B','C','D',
    'E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U',
    'V','W','X','Y','Z');
  nletras<-length(abecedario);

  filas_cerebro<-dim(cerebro)[1];
  columnas_cerebro<-dim(cerebro)[2];
  altura_cerebro<-dim(cerebro)[3];

  abecedario_pesos<-matrix(scan(file="abecedario.txt"),nrow=nletras,
    ncol=altura_cerebro,byrow=TRUE);

  letra_peso<-abecedario_pesos[which(abecedario==letra),];
  matriz_activacion<-matrix(nrow=filas_cerebro,ncol=columnas_cerebro);

  aux<-matrix(nrow=2,ncol=altura_cerebro);
  aux[1,<-letra_peso;
  for(i in 1:filas_cerebro)
  {
    for(j in 1:columnas_cerebro)
    {
      aux[2,<-cerebro[i,j,];
      matriz_activacion[i,j]<-dist(x=aux);
    }
  }
  lista_activacion<-sort(as.vector(matriz_activacion));

  matriz_colores<-matrix(data=rep(0,filas_cerebro*columnas_cerebro),
    nrow=filas_cerebro,ncol=columnas_cerebro);

  izq<-min(lista_activacion);
  rango<-max(lista_activacion)-min(lista_activacion);
  for(i in 1:filas_cerebro)

```

```

{
  for(j in 1:columnas_cerebro)
  {
    if(matriz_activacion[i,j]>=izq+0.75*rango)
    {
      matriz_colores[i,j]<-"gray100";
    }
    if(matriz_activacion[i,j]>izq+0.5*rango &
matriz_activacion[i,j]<izq+0.75*rango)
    {
      matriz_colores[i,j]<-"gray70";
    }
    if(matriz_activacion[i,j]>izq+0.25*rango &
matriz_activacion[i,j]<=izq+0.5*rango)
    {
      matriz_colores[i,j]<-"gray40";
    }
    if(matriz_activacion[i,j]<=izq+0.25*rango)
    {
      matriz_colores[i,j]<-"gray0";
    }
  }
}
barplot(rep(10,columnas_cerebro),border=NA,axes=FALSE,space=0,
col=matriz_colores[1,],

ylim=c(1,filas_cerebro),main=letra);

for(i in seq(from=filas_cerebro-1,to=1,by=-1))
{
  barplot(rep(i,columnas_cerebro),border=NA,axes=FALSE,space=0,
col=matriz_colores[i,],add=TRUE);
}
}

```

```

som1hist<-function(cerebro,letra='a')
{
  abecedario<-c('a','b','c','d','e','f','g','h','i','j','k','l','m',
    'n','o','p','q','r','s','t','u','v','w','x','y','z','A','B','C','D',
    'E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U',
    'V','W','X','Y','Z');
  nletras<-length(abecedario);

  filas_cerebro<-dim(cerebro)[1];
  columnas_cerebro<-dim(cerebro)[2];
  altura_cerebro<-dim(cerebro)[3];

  abecedario_pesos<-matrix(scan(file="abecedario.txt"),nrow=nletras,
    ncol=altura_cerebro,byrow=TRUE);

  letra_peso<-abecedario_pesos[which(abecedario==letra),];
  matriz_activacion<-matrix(nrow=filas_cerebro,ncol=columnas_cerebro);

  aux<-matrix(nrow=2,ncol=altura_cerebro);
  aux[1,<-letra_peso;
  for(i in 1:filas_cerebro)
  {
    for(j in 1:columnas_cerebro)
    {
      aux[2,<-cerebro[i,j,];
      matriz_activacion[i,j]<-dist(x=aux);
    }
  }
  lista_activacion<-sort(as.vector(matriz_activacion));

  matriz_colores<-matrix(data=rep(0,filas_cerebro*columnas_cerebro),
    nrow=filas_cerebro,ncol=columnas_cerebro);

  izq<-min(lista_activacion);
  rango<-max(lista_activacion)-min(lista_activacion);
  for(i in 1:filas_cerebro)

```

```

{
  for(j in 1:columnas_cerebro)
  {
    if(matriz_activacion[i,j]>=izq+0.75*rango)
    {
      matriz_colores[i,j]<-"gray100";
    }
    if(matriz_activacion[i,j]>izq+0.5*rango &
matriz_activacion[i,j]<izq+0.75*rango)
    {
      matriz_colores[i,j]<-"gray70";
    }
    if(matriz_activacion[i,j]>izq+0.25*rango &
matriz_activacion[i,j]<=izq+0.5*rango)
    {
      matriz_colores[i,j]<-"gray40";
    }
    if(matriz_activacion[i,j]<=izq+0.25*rango)
    {
      matriz_colores[i,j]<-"gray0";
    }
  }
}

hist3D(x=seq(from=0,to=filas_cerebro-1,by=1),
y=seq(from=0,to=columnas_cerebro-1,
by=1),z=matriz_activacion,col=m

}

som3<-function(f_cerebro=10,c_cerebro=10,a_cerebro=5,nrep=1000,mw=25,
nw=25,nrepw=150)
{
  abecedario<-c('a','b','c','d','e','f','g','h','i','j','k','l','m',
'n','o','p','q','r','s','t','u','v','w','x','y','z','A','B','C','D',
'E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U',
'V','W','X','Y','Z');
  nletras<-length(abecedario);

```

```

cerebro_letras<-som1(filas_cerebro=f_cerebro,columnas_cerebro=c_cerebro,
altura_cerebro=a_cerebro,nrep=1000);
abecedario_pesos<-matrix(scan(file="abecedario.txt"),nrow=nletras,
ncol=a_cerebro,byrow=TRUE);

palabras<-as.vector(read.table("3lw.txt")$V1);
npalabras<-length(palabras);

palabras_pesos<-matrix(nrow=npalabras,ncol=dim(cerebro_letras)[3]*3);

for(k in 1:npalabras)
{
  palabra<-palabras[k];
  letras_separadas<-unlist(strsplit(palabra,split=""));
  palabra_peso<-vector();
  for(kk in 1:3)
  {
    estimulo<-abecedario_pesos[which(abecedario==letras_separadas[kk]),];
    i0<-1;
    j0<-1;
    neurona_activa<-cerebro_letras[i0,j0,];
    for(i in 1:f_cerebro)
    {
      for(j in 1:c_cerebro)
      {
        celula<-cerebro_letras[i,j,];
        if(norm(x=celula-estimulo,type="2")<norm(x=celula-neurona_activa,
type="2"))
        {
          i0<-i;
          j0<-j;
          neurona_activa<-celula;
        }
      }
    }
  }
  palabra_peso<-c(palabra_peso,neurona_activa);
}

```

```

    }
    palabras_pesos[k,]<-palabra_peso;
  }

write(x=palabras_pesos,file="palabras_pesos.txt");

r0<-0.5*max(mw,nw);
lambda<-nrep/log(r0);
aprendizaje<-0.9;

cerebro_palabras<-array(data=runif(n=mw*nw*a_cerebro*3,min=0,max=1),
dim=c(mw,nw,a_cerebro*3));
for(t in 1:nrepw)
{
  i0<-1;
  j0<-1;
  neurona_activa<-cerebro_palabras[i0,j0,];
  estimulo<-palabras_pesos[sample(x=1:npalabras,size=1),];
  for(i in 1:mw)
  {
    for(j in 1:nw)
    {
      celula<-cerebro_palabras[i,j,];
      if(norm(x=celula-estimulo,type="2")<norm(x=celula-neurona_activa,
type="2"))
      {
        i0<-i;
        j0<-j;
        neurona_activa<-celula;
      }
    }
  }
}

r<-r0*exp(-t/lambda);
aprendizaje<-aprendizaje*exp(-t/lambda);

```

```

for(i in 1:mw)
{
  for(j in 1:nw)
  {
    if(i!=i0 & j!=j0)
    {
      # Distancia a la célula activa
      d<-dist(matrix(c(i0,i,j0,j),2,2));
      entorno<-exp(-(d^2)/(2*(r^2)));
      gamma<-entorno*aprendizaje;

      cerebro_palabras[i,j,<-cerebro_palabras[i,j,]+
      gamma*(estimulo-cerebro_palabras[i0,j0,]);
    }
  }
}

return(cerebro_palabras);
}

som3g<-function(cerebro_palabras,palabra="bdz")
{
  m<-dim(cerebro_palabras)[1];
  n<-dim(cerebro_palabras)[2];
  p<-dim(cerebro_palabras)[3];

  palabras<-as.vector(read.table("3lw.txt")$V1);
  npalabras<-length(palabras);
  palabras_pesos<-matrix(data=scan(file="palabras_pesos.txt"),
  nrow=length(scan(file="palabras_pesos.txt")),ncol=p,byrow=TRUE);

  palabra_peso<-palabras_pesos[which(palabras==palabra),];
  matriz_activacion<-matrix(nrow=m,ncol=n);

  aux<-matrix(nrow=2,ncol=p);
  aux[1,<-palabra_peso;

```

```

for(i in 1:m)
{
  for(j in 1:n)
  {
    aux[2,]<-cerebro_palabras[i,j,];
    matriz_activacion[i,j]<-dist(x=aux);
  }
}
lista_activacion<-sort(as.vector(matriz_activacion));

matriz_colores<-matrix(nrow=m,ncol=n);
for(i in 1:m)
{
  for(j in 1:n)
  {
    if(matriz_activacion[i,j]>=quantile(lista_activacion,0.75))
    {
      matriz_colores[i,j]<-"gray100";
    }
    if(matriz_activacion[i,j]>median(lista_activacion) & matriz_activacion[i,j]<quantile(lista_activacion,0.75))
    {
      matriz_colores[i,j]<-"gray70";
    }
    if(matriz_activacion[i,j]>quantile(lista_activacion,0.25) & matriz_activacion[i,j]<median(lista_activacion))
    {
      matriz_colores[i,j]<-"gray40";
    }
    if(matriz_activacion[i,j]<=quantile(lista_activacion,0.25))
    {
      matriz_colores[i,j]<-"gray0";
    }
  }
}

barplot(rep(10,n),border=NA,axes=FALSE,space=0,col=matriz_colores[1,],
ylim=c(1,m),main=palabra);

```



```

for(i in seq(from=m-1,to=1,by=-1))
{
  barplot(rep(i,n),border=NA,axes=FALSE,space=0,col=matriz_colores[i,],
    add=TRUE);
}
}

```

```

som3hist<-function(cerebro_palabras,palabra='bdz')
{
  m<-dim(cerebro_palabras)[1];
  n<-dim(cerebro_palabras)[2];
  p<-dim(cerebro_palabras)[3];

  palabras<-as.vector(read.table("3lw.txt")$V1);
  npalabras<-length(palabras);
  palabras_pesos<-matrix(data=scan(file="palabras_pesos.txt"),
    nrow=length(scan(file="palabras_pesos.txt")),ncol=p,byrow=TRUE);

  palabra_peso<-palabras_pesos[which(palabras==palabra),];
  matriz_activacion<-matrix(nrow=m,ncol=n);

  aux<-matrix(nrow=2,ncol=p);
  aux[1,<-palabra_peso;
  for(i in 1:m)
  {
    for(j in 1:n)
    {
      aux[2,<-cerebro_palabras[i,j,];
      matriz_activacion[i,j]<-dist(x=aux);
    }
  }
  lista_activacion<-sort(as.vector(matriz_activacion));

  izq<-min(lista_activacion);
  rango<-max(lista_activacion)-min(lista_activacion);
  matriz_colores<-matrix(nrow=m,ncol=n);

```

```

for(i in 1:m)
{
  for(j in 1:n)
  {
    if(matriz_activacion[i,j]>=izq+0.75*rango)
    {
      matriz_colores[i,j]<-"gray100";
    }
    if(matriz_activacion[i,j]>izq+0.5*rango &
matriz_activacion[i,j]<izq+0.75*rango)
    {
      matriz_colores[i,j]<-"gray70";
    }
    if(matriz_activacion[i,j]>izq+0.25*rango &
matriz_activacion[i,j]<=izq+0.5*rango)
    {
      matriz_colores[i,j]<-"gray40";
    }
    if(matriz_activacion[i,j]<=izq+0.25*rango)
    {
      matriz_colores[i,j]<-"gray0";
    }
  }
}

hist3D(x=seq(from=0,to=m-1,by=1),y=seq(from=0,to=n-1,by=1),
z=matriz_activacion,col=matriz_colores);
}

```

## 4.4 Experimento 1. Generación aleatoria de los pesos de los estímulos.

### 4.4.1 Mapa autoorganizado: letras

#### 4.4.1.1 Comparación de parejas minúscula / mayúscula

Se presentan las *ALI* emparejadas minúscula / minúscula y mostrando tres casuísticas diferenciadas: 1) Similitud baja en el par minúscula-mayúscula (por ejemplo, “a” frente a “A”), 2) Similitud moderada en el par minúscula-mayúscula (por ejemplo, “e” frente a “E”), y 3) Similitud alta en el par minúscula-mayúscula (por ejemplo, “c” frente a “C”).

En cada situación se muestran en primer lugar gráficos planos con el grado de activación frente al correspondiente estímulo, y en segundo lugar el mismo tipo de gráfico pero en modo 3D.

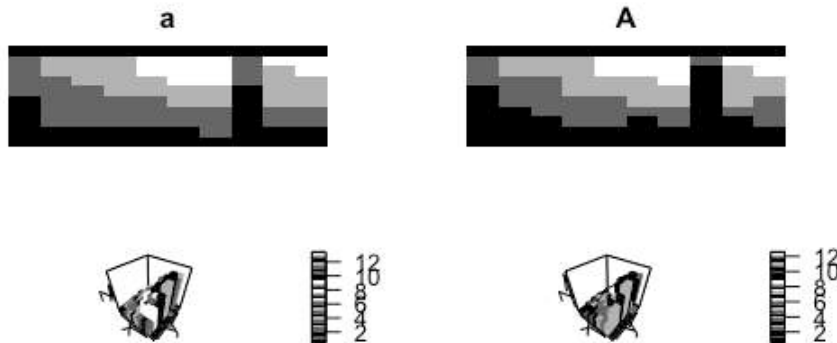


Figura 4.1: Aprendizaje de letras. Similitud baja.

Partimos de un cerebro aleatorio sin entrenar en que los pesos de las neuronas se distribuyen  $Un(0,1)$ . La distribución de las letras es igualmente  $Un(0,1)$ . Al final del entrenamiento se obtienen abstracciones similares *mayúscula-minúscula* para el mismo concepto. Resulta llamativo que los patrones de activación neuronal son cercanos incluso entre conceptos diferentes. Esto sería realmente porque los pesos de partida son similares en todos los casos.

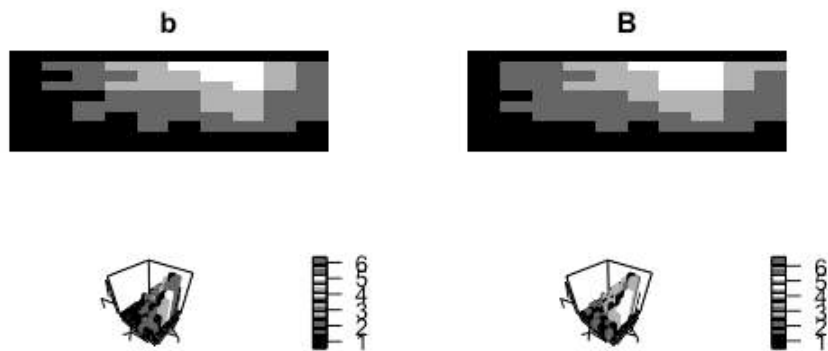


Figura 4.2: Aprendizaje de letras. Similitud moderada.

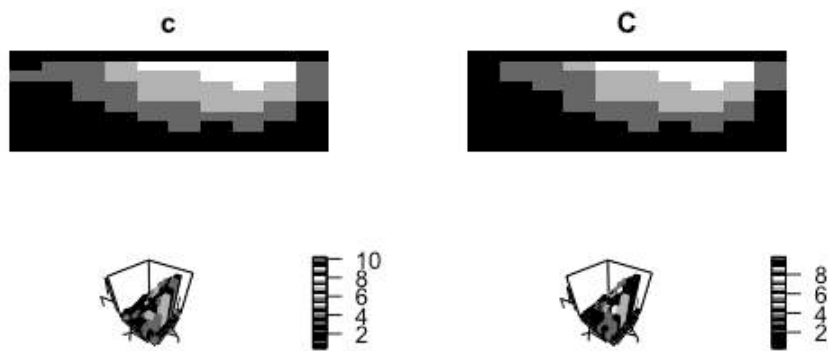


Figura 4.3: Aprendizaje de letras. Alta similitud.

4.4.1.2 Mapa de letras entrenado. Resultado global.

y	c	N	S	N	S	E	c	c	p
r	r	r	r	r	r	r	r	r	L
r	r	r	r	r	r	r	r	r	W
j	r	r	r	r	r	r	r	r	l
r	r	r	r	r	r	r	r	r	m
Q	r	j	r	r	r	r	r	r	i
r	r	r	r	r	r	r	r	r	e
l	u	r	r	r	r	r	r	r	j
r	j	j	r	r	r	j	r	r	L
z	r	h	r	L	r	j	j	j	V

Figura 4.4: Método 1. Mapa de letras entrenado.

Letra	Peso
a	1.35572032947845
b	1.15123533755411
c	1.53316526103694
d	1.09101083990202
e	1.37687303146134
f	1.08341042551265
g	1.22092328682103
h	1.32488836762892
i	1.17878873004156
j	1.70415608478826
k	1.30279245304233
l	1.0754904876822
m	1.43515242205028
n	1.25845980148416
o	0.917047630464562
p	0.876601883219779
q	1.07130892043923

Letra	Peso
r	1.35595804862017
s	1.29912572734248
t	1.18793095779107
u	1.39096803077952
v	0.977445182437697
w	1.22618553489138
x	1.40375674519331
y	1.61854811705572
z	1.26798455533933
A	1.01961797317149
B	1.37982574875573
C	0.756735703874244
D	0.991908114027235
E	1.24664682352277
F	1.27822068734175
G	0.711040297258882
H	0.960995790190467
I	1.6067473744483
J	1.81048493202033
K	1.49264547690329
L	1.07587179654232
M	1.0850104545136
N	1.60215287165878
O	1.40065926787688
P	0.829435602388653
Q	1.44558026222341
R	1.23632894370649
S	1.44045202804461
T	1.69071553741436
U	1.89054291123909
V	1.37131262034336
W	1.04812783197847
X	1.02920320620702
Y	1.42849917857086

Letra	Peso
Z	0.640410329058221

Puede verse que conceptos iguales no tienen necesariamente pesos similares. A la vez conceptos muy diferentes en su representación gráfica pueden tener pesos similares en norma.

Esta situación justifica los métodos 2 y 3 en que se intenta generar pesos para las letras más cercanos a su representación gráfica (método 2) llegando a incorporar una parte fonética.

#### **4.4.2 Mapa autoorganizado: palabras**

##### **4.4.2.1 Comparación de parejas minúscula / mayúscula**

Una vez completado el aprendizaje de letras, independientemente del grado de similitud entre pares de palabras con baja o alta similitud minúscula-mayúscula, se obtienen patrones neuronales similares.

##### **4.4.2.2 Mapa de palabras entrenado. Resultado global.**

### **4.5 Experimento 2. Generación de los pesos los estímulos por representación visual.**

#### **4.5.1 Mapa autoorganizado: letras**

##### **4.5.1.1 Comparación de parejas minúscula / mayúscula**

Se presentan las *ALI* emparejadas minúscula / minúscula y mostrando tres casuísticas diferenciadas: 1) Similitud baja minúscula-mayúscula, 2) Similitud moderada minúscula-mayúscula, y 3) Similitud alta minúscula-mayúscula.

En cada situación se muestran en primer lugar gráficos planos con el grado de activación frente al correspondiente estímulo, y en segundo lugar el mismo tipo de gráfico pero en modo 3D.

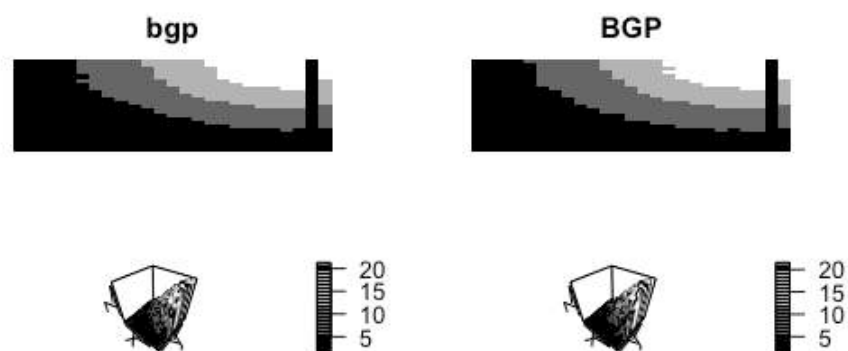


Figura 4.5: Aprendizaje de palabras. Comparación de representaciones. Poca similitud.

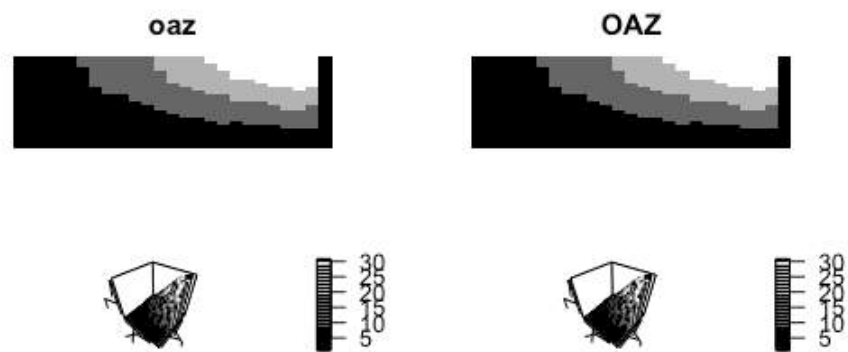


Figura 4.6: Aprendizaje de palabras. Comparación de representaciones. Similitud moderada.



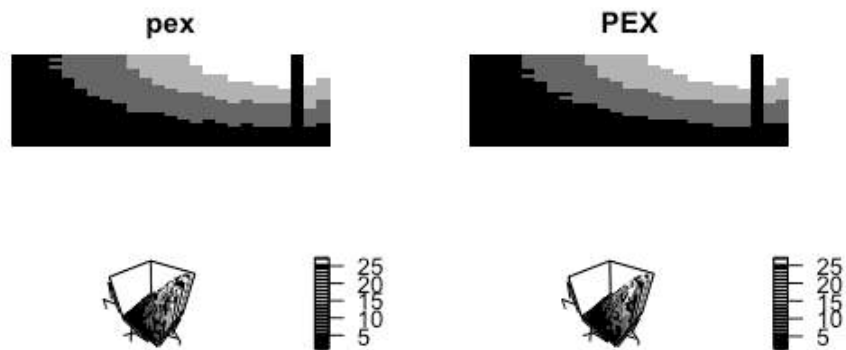


Figura 4.7: Aprendizaje de palabras. Comparación de representaciones. Alta similitud.

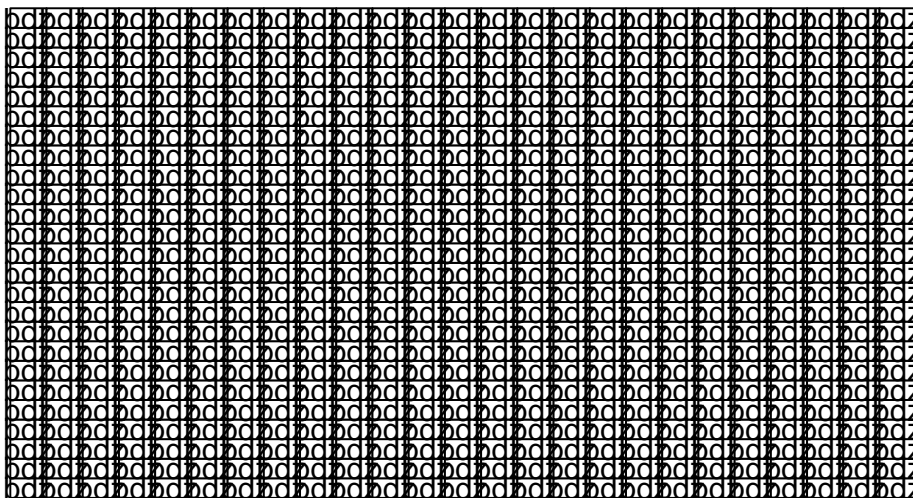


Figura 4.8: Método 1. Mapa de palabras entrenado.

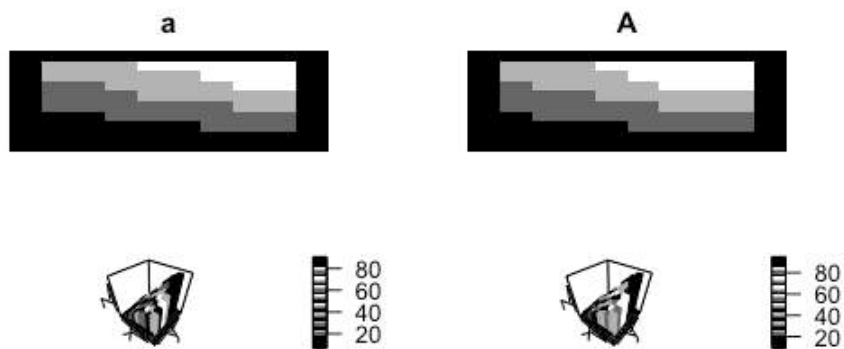


Figura 4.9: Aprendizaje de letras. Similitud baja.

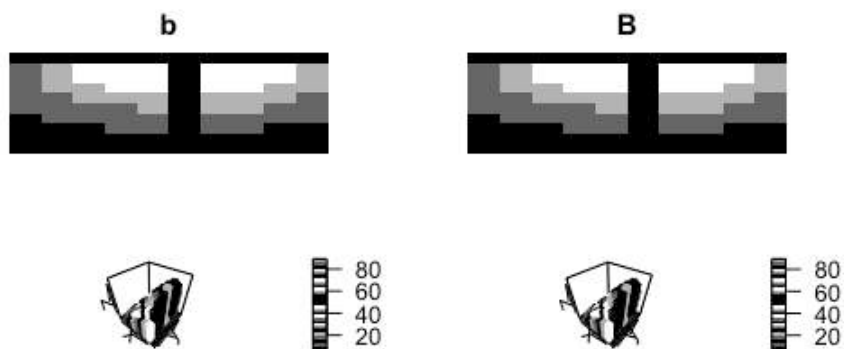


Figura 4.10: Aprendizaje de letras. Similitud moderada.

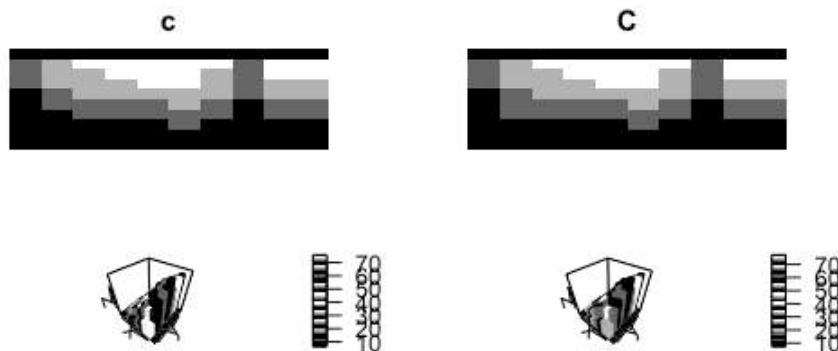


Figura 4.11: Aprendizaje de letras. Alta similitud.

Partiendo de vectores pesos de las letras obtenidos directamente de la lectura del patrón visual de su representación la similitud minúscula-mayúscula es mucho mayor con representaciones visuales más cercanas.

#### 4.5.1.2 Mapa de letras entrenado. Resultado global.

### 4.5.2 Mapa autoorganizado: palabras

#### 4.5.2.1 Comparación de parejas minúscula / mayúscula

Tal y como ya se vió en el método 1, los pesos de palabras parten de un cerebro entrenado previamente para las letras. No se aprecia gran mejora en los patrones de activación en palabras en este caso. La mejora es a nivel de letras.

h	h	h	h	h	h	h	h	h	K
o	o	a	o	a	o	o	o	a	K
h	h	h	h	h	h	h	h	h	w
h	h	h	h	h	h	h	h	h	Q
h	h	h	h	h	h	h	h	h	l
h	o	h	h	h	h	h	h	h	a
h	q	h	h	h	h	h	h	h	o
h	h	o	h	h	h	h	h	h	w
h	h	o	h	h	h	h	h	h	G
a	o	h	o	a	h	h	h	h	a

Figura 4.12: Método 2. Mapa de letras entrenado.

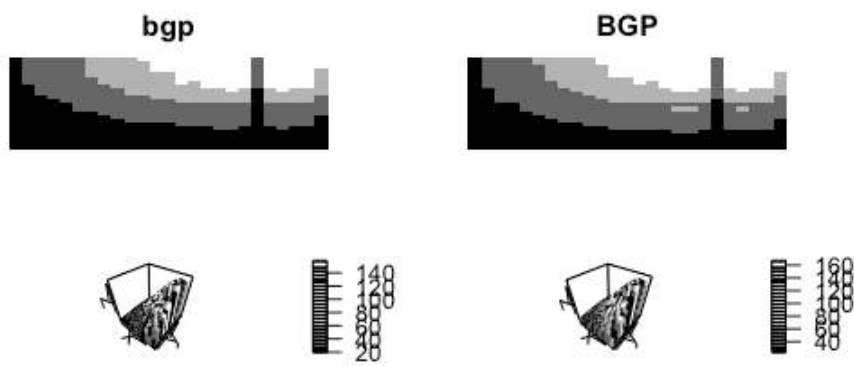


Figura 4.13: Aprendizaje de palabras. Comparación de representaciones. Poca similitud.

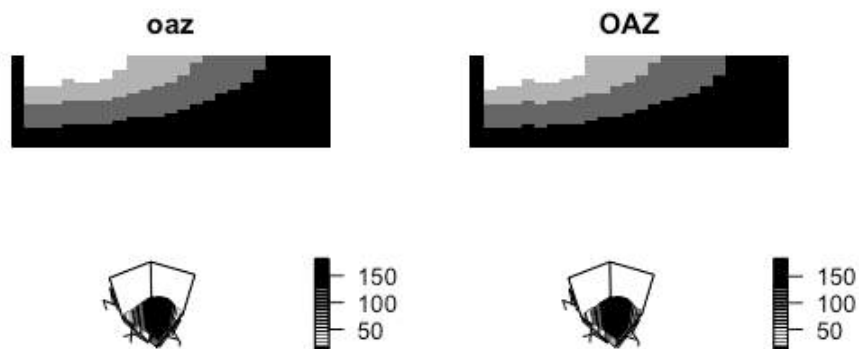


Figura 4.14: Aprendizaje de palabras. Comparación de representaciones. Similitud moderada.

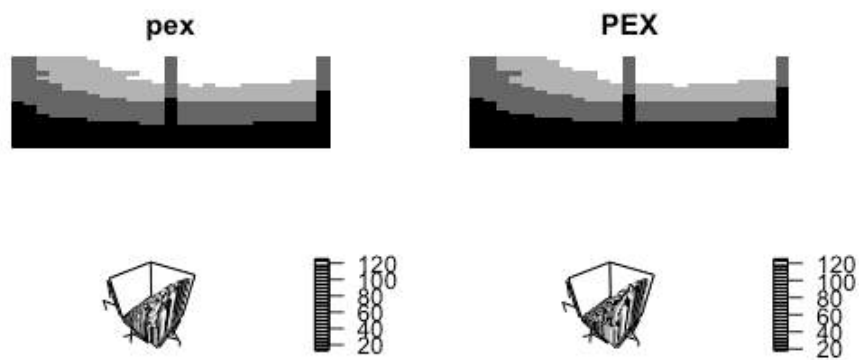


Figura 4.15: Aprendizaje de palabras. Comparación de representaciones. Alta similitud.

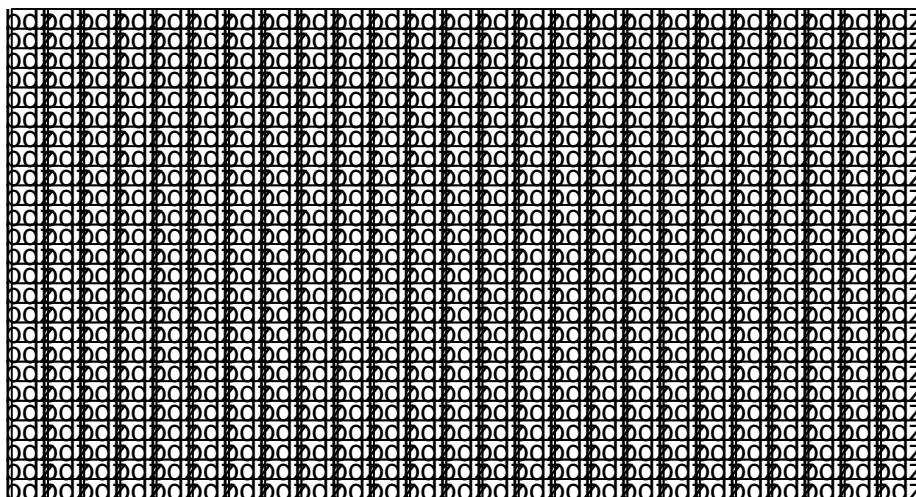


Figura 4.16: Método 2. Mapa de palabras entrenado.

#### 4.5.2.2 Mapa de palabras entrenado. Resultado global.

### 4.6 Método 3. Generación de los pesos de los pesos de los estímulos por representación visual. Incorporación del elemento fonético.

#### 4.6.1 Mapa autoorganizado: letras

##### 4.6.1.1 Comparación de parejas minúscula / mayúscula

Se presentan las *ALI* emparejadas minúscula / minúscula y mostrando tres casuísticas diferenciadas: 1) Similitud baja minúscula-mayúscula, 2) Similitud moderada minúscula-mayúscula, y 3) Similitud alta minúscula-mayúscula.

En cada situación se muestran en primer lugar gráficos planos con el grado de activación frente al correspondiente estímulo, y en segundo lugar el mismo tipo de gráfico pero en modo 3D.

La representación minúscula-mayúscula es indudablemente mayor al final del entrenamiento de palabras cuando los pesos de partida de las letras incorporan

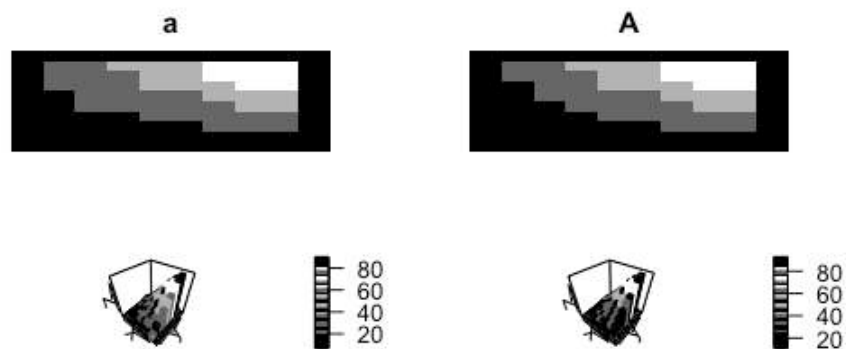


Figura 4.17: Aprendizaje de letras. Similitud baja.

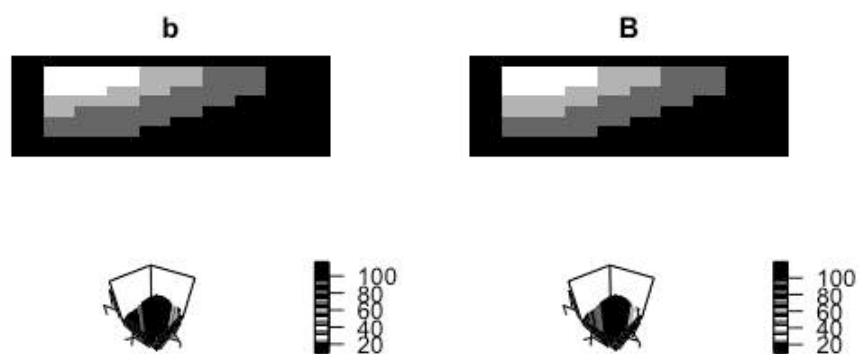


Figura 4.18: Aprendizaje de letras. Similitud moderada.

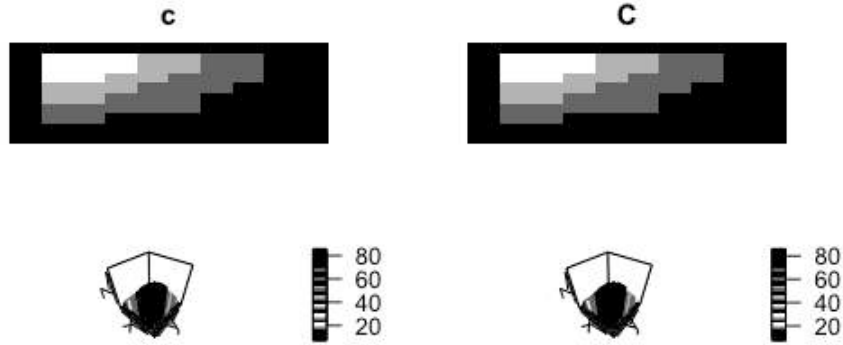


Figura 4.19: Aprendizaje de letras. Alta similitud.

la capa fonética.

#### 4.6.1.2 Mapa de letras entrenado. Resultado global.

### 4.6.2 Mapa autoorganizado: palabras

#### 4.6.2.1 Comparación de parejas minúscula / mayúscula

No se observan diferencias significativas entre los pares minúscula-mayúscula cuando se incorpora una capa fonética.

#### 4.6.2.2 Mapa de palabras entrenado. Resultado global.

## 4.7 Error

Se define la *Precisión de la Proyección* como una medida de evaluar la precisión del mapa entrenado. Sean  $N$  el número de estímulos que sirven para entrenar el mapa,  $x_i$  cada uno de los pesos de los diferentes estímulos y  $m_c$  el peso de la neurona más cercano al correspondiente estímulo.

$$\epsilon_p = \frac{1}{N} \sum_{i=1}^N \| x_i - m_c \|$$



X	V	Z	D	D	D	D	D	x	a
X	x	x	x	x	x	x	x	x	X
Z	x	x	x	x	x	x	x	x	Z
x	x	x	x	x	x	x	x	x	a
X	X	x	x	x	x	x	x	x	Z
X	X	x	x	x	x	x	x	x	Y
X	X	x	x	x	x	x	x	x	Z
X	X	x	x	x	x	x	x	x	a
X	X	X	x	D	x	x	x	x	V
X	X	X	X	X	D	x	x	D	X

Figura 4.20: Método 3. Mapa de letras entrenado.

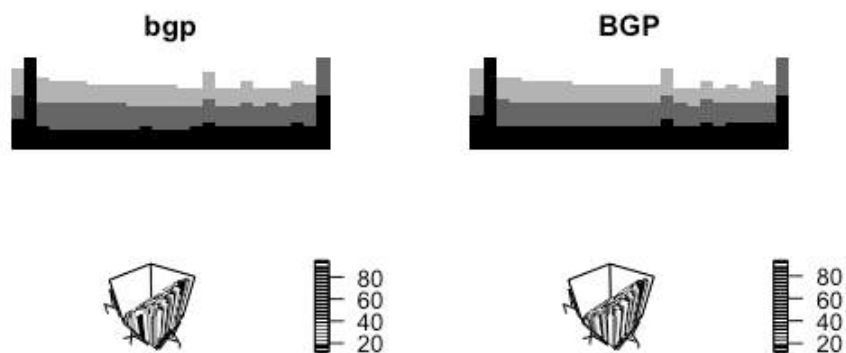


Figura 4.21: Aprendizaje de palabras. Comparación de representaciones. Poca similitud.

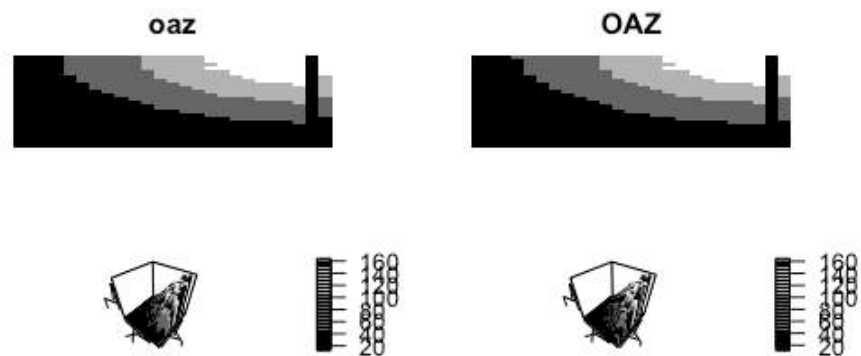


Figura 4.22: Aprendizaje de palabras. Comparación de representaciones. Similitud moderada

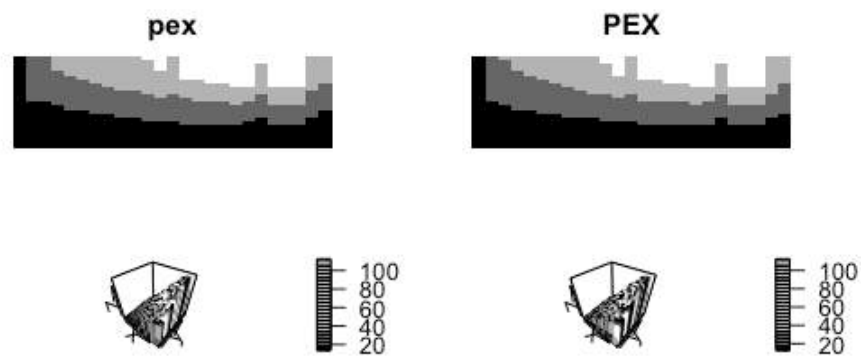


Figura 4.23: Aprendizaje de palabras. Comparación de representaciones. Alta similitud.

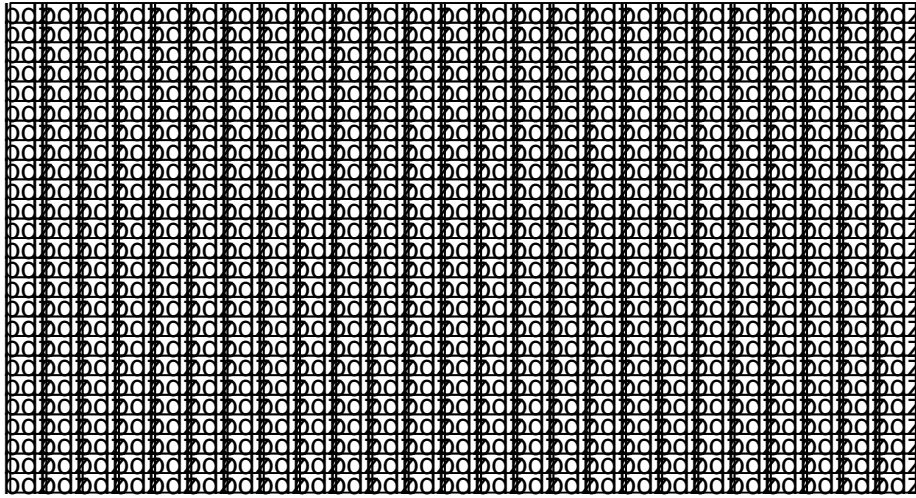


Figura 4.24: Método 3. Mapa de palabras entrenado.

El Error de *Precisión de la Proyección* para cada uno de los tres métodos se estudia en una sección propia en lugar de en cada una de las anteriores para poder comparar los errores comparativos de los diferentes métodos.

#### 4.7.1 Letras

Analizamos en esta sección la Precisión de la Proyección en el aprendizaje de letras para cada uno de los tres métodos desarrollados en las secciones anteriores. Cada gráfico muestra la evolución del aprendizaje con *1000* iteraciones, aunque por una cuestión visual sólo se mostrarán las iteraciones *1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 50, 100, 200, 500 y 1000*.

La diferencia en los rangos de valores del error es debida a la creciente dimensión de los vectores peso en los diferentes métodos. Lo relevante es que el error se estabiliza en todos los casos a partir de la iteración *50*, sobre un total de *1500* iteraciones.

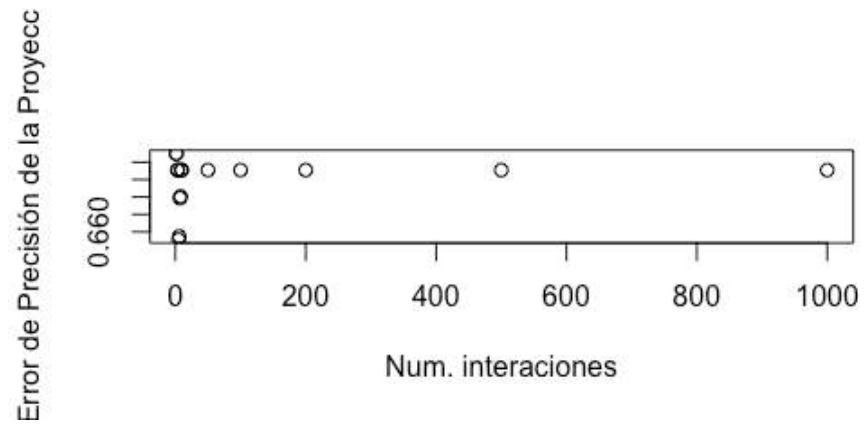


Figura 4.25: Aprendizaje de letras. Error de Precisión de la Proyección. Método 1.

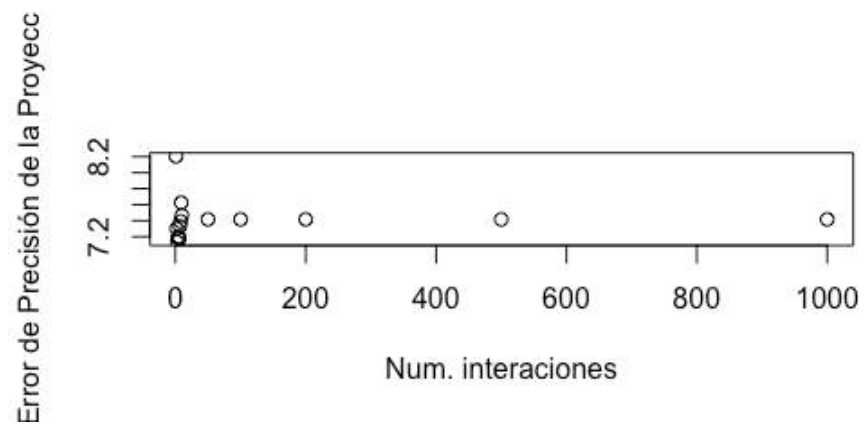


Figura 4.26: Aprendizaje de letras. Error de Precisión de la Proyección. Método 2.

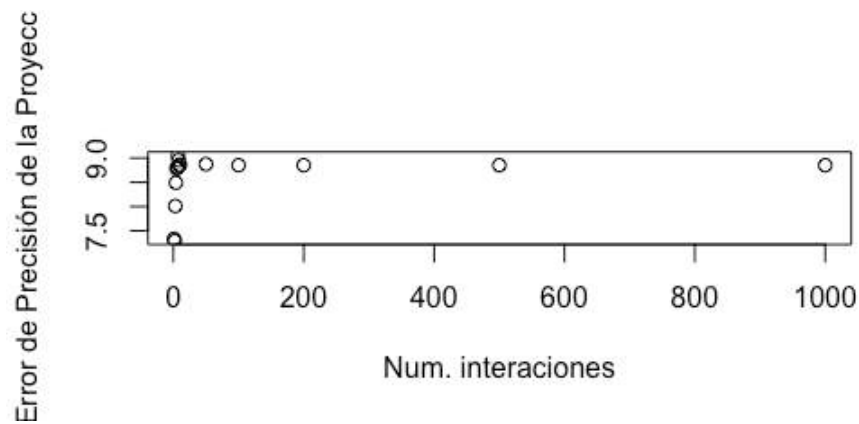


Figura 4.27: Aprendizaje de letras. Error de Precisión de la Proyección. Método 3.

#### 4.7.2 Palabras

Analizamos en esta sección la Precisión de la Proyección en el aprendizaje de palabras para cada uno de los tres métodos desarrollados en las secciones anteriores. Cada gráfico muestra la evolución del aprendizaje con 150 iteraciones, aunque por una cuestión visual sólo se mostrarán las iteraciones 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 50, 150.

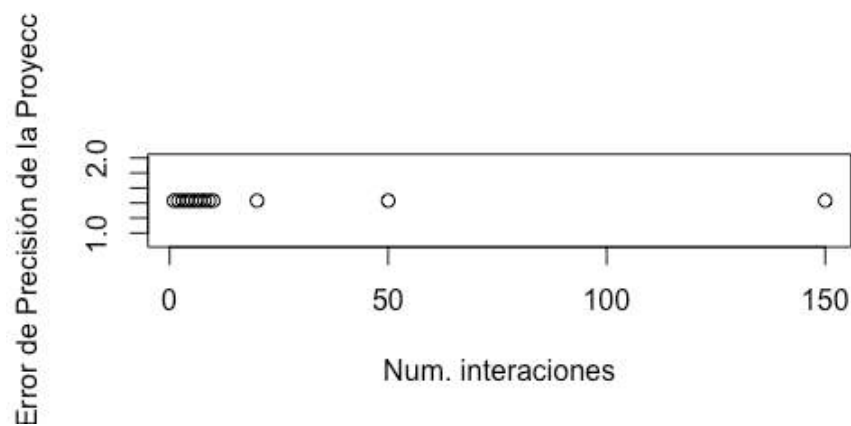


Figura 4.28: Aprendizaje de letras. Error de Precisión de la Proyección. Método 1.

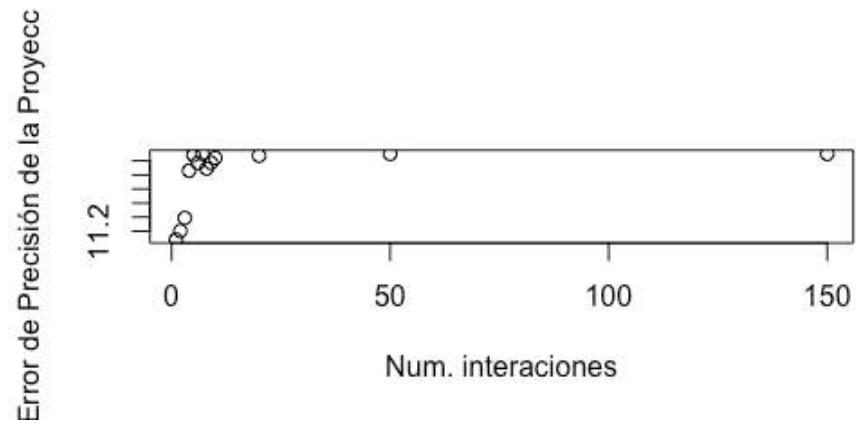


Figura 4.29: Aprendizaje de letras. Error de Precisión de la Proyección. Método 2.

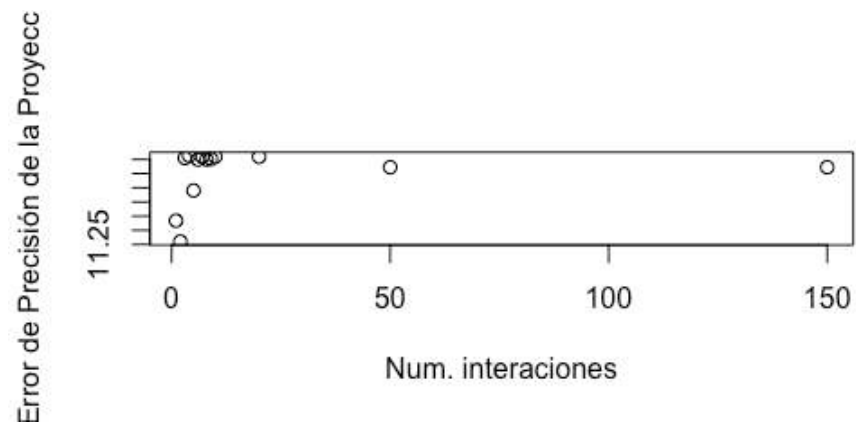


Figura 4.30: Aprendizaje de letras. Error de Precisión de la Proyección. Método 3.

Al igual que en el aprendizaje de letras, el rango de valores del error es creciente entre los diferentes métodos debido a la dimensión de los vectores peso en cada caso. Sin embargo, es reseñable como el error se estabiliza casi desde las iteraciones iniciales en el primer método y algo más tardíamente en los métodos que le siguen.

## 4.8 Conclusiones

En el aprendizaje de letras se parte de una base o cerebro sin entrenar con neuronas  $Un(0,1)$ . La diferencia entre las tres opciones planteadas se basa en la forma de tomar los vectores peso de las letras o estímulos. Ya se explicó inicialmente que en un primer acercamiento al método se parte de vectores peso para los estímulos basados en la misma distribución  $Un(0,1)$ . En opciones posteriores se toman los estímulos basándose en la lectura de la representación gráfica y la incorporación de una capa fonética. Las diferentes formas de tomar los vectores pesos de los estímulos hacen que con el mismo número de iteraciones de entrenamiento se alcance mayor grado de similitud a nivel de activación neuronal entre los pares minúscula-mayúscula independientemente de su similitud visual. En lo referente al aprendizaje de palabras, se ha podido ver que la forma de determinar en el aprendizaje de letras los vectores peso de los diferentes estímulos no es determinante en el grado de activación neuronal para las palabras una vez completado su aprendizaje. Esto es debido a que los vectores peso de las palabras se basan en la concatenación de los vectores peso correspondientes a la neurona de mayor activación frente a cada letra de las que componen la palabra estudiada. De este modo, el aprendizaje de las palabras se basa en el de las letras. Es decir, el método de determinación de los vectores peso para los diferentes estímulos letras es importante para alcanzar el mismo *ALI* en las parejas minúscula-mayúscula. Sin embargo, dicho método no es clave para el aprendizaje de las palabras una vez entrenado el cerebro para el aprendizaje de las letras.





## Parte III

# Tratamiento de outliers en Tiempos de Respuesta de reconocimiento de palabras



## Capítulo 5

# Generación datos en tiempos de respuesta de reconocimiento de palabras con presencia de outliers y posibilidades de transformación de los mismos

### 5.1 Introducción

La recogida y análisis posterior de Tiempos de Respuesta (en adelante *TR*) en reconocimiento de palabras ha sido durante las últimas décadas el ámbito de trabajo de la *Psicología cognitiva*.

En la recogida de estos datos, independientemente de la disciplina de estudio,

es habitual encontrar datos extremos o *outliers*. Con frecuencia, el preprocesado de los conjuntos de datos, basado en diferentes transformaciones de los mismos y con el objeto de reducir la influencia de dichos datos extremos, ha consistido en diferentes opciones:

- $\bar{TR} \pm 2\tilde{TR}$ ,
- $\log(TR)$ ,
- $\frac{1}{TR}$ ,
- etc.

Una vez completado este preprocesamiento, el análisis se suele hacer empleando la técnica clásica *t-test* (modelo lineal). Concretamente, un análisis de la potencia de los diferentes métodos de tratamiento de *outliers* puede verse en el trabajo de Ratcliff (1993).

Más recientemente, dadas las limitaciones estructurales de un análisis puramente lineal, se han empleado *Modelos Lineales Generalizados Mixtos*. Pueden verse lo planteado en los trabajos de Baayen, Davidson & Bates (2008) y Barra, Levyb, Scheepersa & Tilyc (2013).

En base al trabajo de Ratcliff (1993), la transformación que mejores resultados ha ofrecido con el análisis posterior basado en un modelo lineal, es  $\frac{1}{TR}$ .

Empleando, al igual que en Ratcliff (1993), una distribución *ExGauss*( $\mu = 400, \sigma = 40, \tau = 200$ ) y tomando una muestra de 1000 items / lecturas, representaremos la muestra aleatoria con las cuatro posibles transformaciones  $TR$  (No cut o sin transformación),  $\log TR$ ,  $\frac{1}{TR}$  y  $\frac{1}{TR^2}$ .

Como puede apreciarse, la transformación  $\frac{1}{TR^2}$  ofrece una forma similar a  $\frac{1}{TR}$  pero en un rango menor y con una curvatura menos pronunciada en los valores centrales, entre los cuartiles  $Q_1$  y  $Q_3$ . Esto justificaría en un primer momento el análisis de esta propuesta de transformación  $\frac{1}{TR^2}$ . En concreto, para poder aplicar un *Modelo Generalizado Lineal Mixto* es necesario que los datos iniciales se distribuyan de forma aproximadamente normal. Como puede apreciarse la forma de los gráficos  $\frac{1}{TR}$  y  $\frac{1}{TR^2}$  es similar y el planteamiento de un análisis de este tipo sería razonable.

En una primera aproximación, en una muestra de 1 sujeto y 1000 items de tipo *ExGauss*( $\mu = 400, \sigma = 40, \tau = 200$ ) veamos la representación de la función de distribución de los datos sin transformación alguna y con las diferentes posibilidades  $\frac{1}{TR}$  y  $\frac{1}{TR^2}$ .

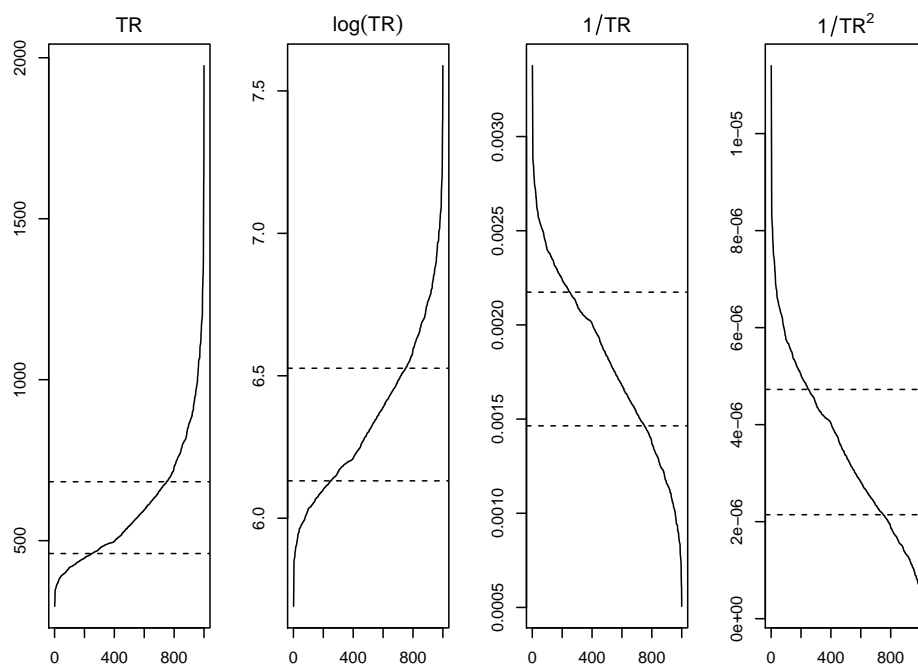


Figura 5.1: Transformaciones de datos

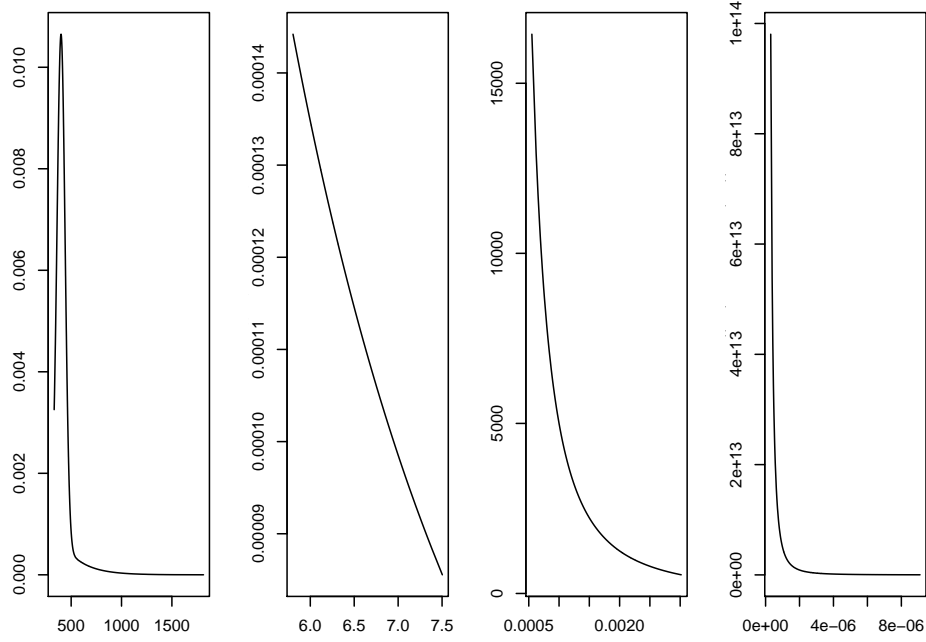


Figura 5.2: Transformaciones de datos. Función de densidad

Conocidas las funciones de densidad de la distribución  $N(\mu = 400, \sigma = 40)$  y  $Ex(\tau = 200)$  se puede calcular la función de densidad de la distribución  $ExGauss(\mu = 400, \sigma = 40, \tau = 200)$  como suma de las funciones de densidad de las dos primeras.

$$f(t) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(t-\mu)^2}{2\sigma^2}} + \tau e^{-\tau t}$$

Dada una transformación  $t' = h(t)$  se calcula la función de densidad de la transformación como  $g(t') = f(t) \left| \frac{dt'}{dt} \right|$ . Calculamos entonces la función de densidad correspondiente a cada transformación.

$$t' = \frac{1}{t} \rightarrow g(t') = \left( \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(\frac{1}{t'} - \mu)^2}{2\sigma^2}} + \tau e^{-\tau \frac{1}{t'}} \right) \frac{1}{t'^2}$$

$$t' = \frac{1}{t^2} \rightarrow g(t') = \left( \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(\frac{1}{\sqrt{t'}} - \mu)^2}{2\sigma^2}} + \tau e^{-\tau \frac{1}{\sqrt{t'}}} \right) \frac{1}{\sqrt{t'^5}}$$

	Sin outliers	10 % outliers
No cut	0.2799550	0.7298260
log(TR)	0.0296119	0.0664722
Inv	0.1150513	0.3323024
Inv2	0.2067944	0.5077276

La dispersión en el caso sin outliers siempre es inferior a la dispersión para el caso con un 10 % de outliers para cada una de las cuatro transformaciones consideradas. Se ha empleado el *Coficiente de dispersión de Pearson*,

$$CV = \frac{\sigma_X}{\bar{X}}$$

, siendo  $X = \{x_i\}$  el conjunto de datos (o sus transformaciones) y  $\bar{X}$  la media del conjunto de datos (o sus transformaciones).

## 5.2 Planteamiento del problema y Modelo Lineal Generalizado Mixto

Consideremos 2 condiciones,  $n_S$  sujetos y  $n_I$  items. Bajo la primera condición pueden por tanto considerarse  $n_S$  muestras de  $n_I$  items independientes e idénticamente distribuidas en base a una distribución *ExGauss* de parámetros  $\mu$ ,  $\sigma$  y  $\tau$ .

Si los  $n_S$  sujetos se someten a los mismos  $n_I$  items debemos tener en cuenta la variabilidad entre items dentro de un mismo sujeto y también la variabilidad entre sujetos para el mismo item. Particularmente, partiendo de *ExGauss*( $\mu, \sigma, \tau$ ), para cada sujeto  $\mu \sim \text{ExGauss}(\mu, \sigma, \tau)$ ,  $\sigma \sim N(\sigma, \frac{\sigma}{2})$  y  $\tau \sim N(\tau, \frac{\tau}{2})$ .

Las  $n_S$  sujetos muestras del segundo sujeto emplean efectos sobre los tres parámetros de la distribución *ExGauss* original dados por  $\mu_{ef}$ ,  $\sigma_{ef}$  y  $\tau_{ef}$ .

De este modo, podemos considerar cada *TR* analizado como perteneciente a un posible sujeto, item y condición. El análisis de datos ya no es por tanto convencional, como en Ratcliff (1993) sino que emplearemos un *Modelo Lineal de Efectos Mixtos*, similarmente a Baayen et al. (2008) y Barr et al (2013).

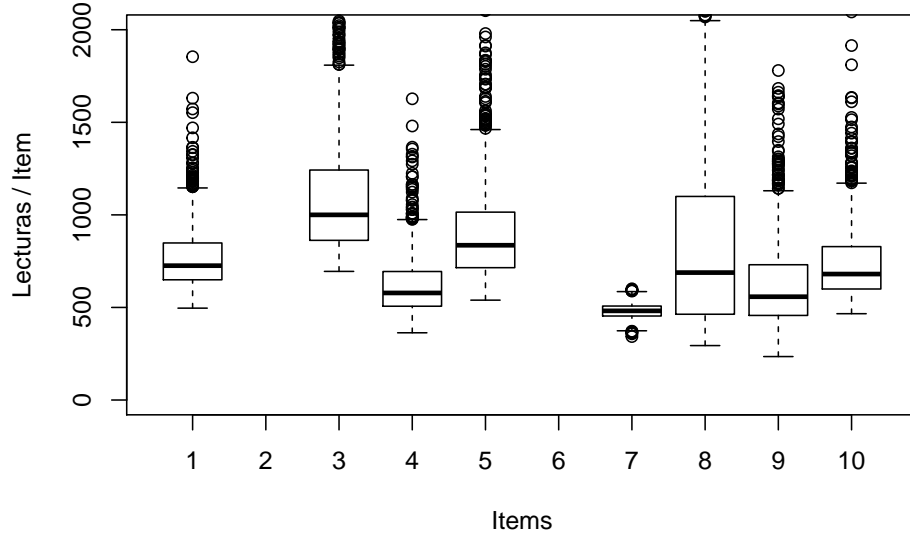


Figura 5.3: Distribución ExGauss de 10 items

El análisis computacional se hará empleando el software estadístico **R-Project** (2006) , embebido dentro de su *FrontEnd R-studio* (2015), que se ha empleado para la redacción del presente documento, conjuntamente con el sistema de procesamiento de textos **TeX** (1984). La librería particular empleada es **lme4** (2015), de uso específico para Modelos Generalizados Lineales Mixtos. Las múltiples simulaciones estadísticas que han permitido generar las muestras respectivas y analizar los correspondientes resultados se han ejecutado en el entorno del *Supercomputador Lluís Vives* de la *Universitat de València*, cuya información detallada puede consultarse en la web del *Centro de Cálculo* de la *Universitat de València* («WIKI del Servicio de Informática SiUV», 2008).

Las cuatro transformaciones posibles de un *TR* son:  $t$  (sin transformación / No cut),  $\log(t)$  (logaritmo neperiano),  $\frac{1}{t}$  (inversa) y  $\frac{1}{t^2}$  (inversa de la potencia 2 de la inversa). Estas funciones transformación son positivas y crecientes si  $t > 0$  y además

$$\frac{1}{t^2} < \frac{1}{t} < \log(t) < t, \forall t \in ]1, +\infty[$$

Matemáticamente, consideraremos los *TR* generados en base a una distribución  $ExGauss(\mu, \sigma, \tau)$  y expresados en *ms*. Sean  $\mu$ ,  $\sigma$  y  $\tau$  los parámetros que definen



la media, desviación típica y cola de la distribución ExGauss  $T$  generada como suma de las distribuciones  $Ex(\tau)$  y  $N(\mu, \sigma)$ . Conocidas las distribuciones de ambas, la función de densidad de la *ExGauss* resultante es de la forma

$$f(t|\mu, \sigma, \tau) = \tau \hat{u} e^{-\tau \hat{u} t} + \frac{1}{\sigma \hat{u} \sqrt{2 \hat{u} \pi}} \hat{u} e^{-\frac{1}{2} \hat{u} (\frac{t-\mu}{\sigma})^2}, t > 0$$

. Por el mismo motivo, podemos obtener la esperanza y varianza de la distribución ExGauss de los  $TR$ .

$$E(T) = \frac{1}{\tau} + \mu, Var(T) = \frac{1}{\tau^2} + \sigma^2, t > 0$$

Particularmente, la distribución *ExGauss* básica de los experimentos que siguen tiene por parámetros  $\mu = 400$ ,  $\sigma = 40$  y  $\tau = 200$ .

### 5.3 Generación de los datos

1. Sea la distribución base *ExGauss*( $\mu, \sigma, \tau$ ). Cada uno de los  $n_S$  sujetos será sometido a  $n_I$  items.
2. Se genera en la distribución dada un conjunto de  $n_I$  medias base que representarán la media de cada item, sean estas  $(\mu_1, \mu_2, \dots, \mu_n)$ .
3. Las  $n_S$  lecturas del  $j$ -ésimo sujeto,  $j = 1, 2, \dots, n$  se obtienen como un muestreo *ExGauss*( $\mu = \mu_j, \sigma, \tau$ ).
4. Los puntos 1 al 3 se refieren a la primera condición. Veamos como generar los datos de la segunda condición. Sean  $\mu_{ef}$ ,  $\sigma_{ef}$  y  $\tau_{ef}$  los efectos aplicados sobre  $\mu$ ,  $\sigma$  y/o  $\tau$ . Repetimos los puntos del 1 al 3 con una distribución *ExGauss*( $\mu + \mu_{ef}, \sigma + \sigma_{ef}, \tau + \tau_{ef}$ ). En este caso los mismos  $n_S$  sujetos se someterán a  $n_I$  nuevos estímulos con las correspondientes variaciones en los parámetros de la distribución.

### 5.4 Distribución de datos

Antes de entrar en un análisis particular de la dispersión de los datos plantearemos una serie de representaciones gráficas de los mismos. Los datos se distribuyen, como será habitual en los siguientes apartados según una distribución base *ExGauss*( $\mu = 400, \sigma = 40, \tau = 200$ ). En cada gráfico se comparan los

datos sin aplicar efecto (línea sólida) y aplicando efecto (línea punteada) sobre alguno de los parámetros  $\mu$ ,  $\sigma$  o  $\tau$ . Se pretende comparar la distribución de datos por lo que no se entra en pormenores de número de sujetos ni número de items sino que los datos se basan en un supersujeto de 1000 lecturas.

### 5.4.1 Efectos sobre la media

Los efectos sobre la media  $\mu_{ef}$  toman posibles valores  $0$ ,  $20$ ,  $50$  y  $100$  ms. (cada fila) y las posibles transformaciones se muestran en cada columna.

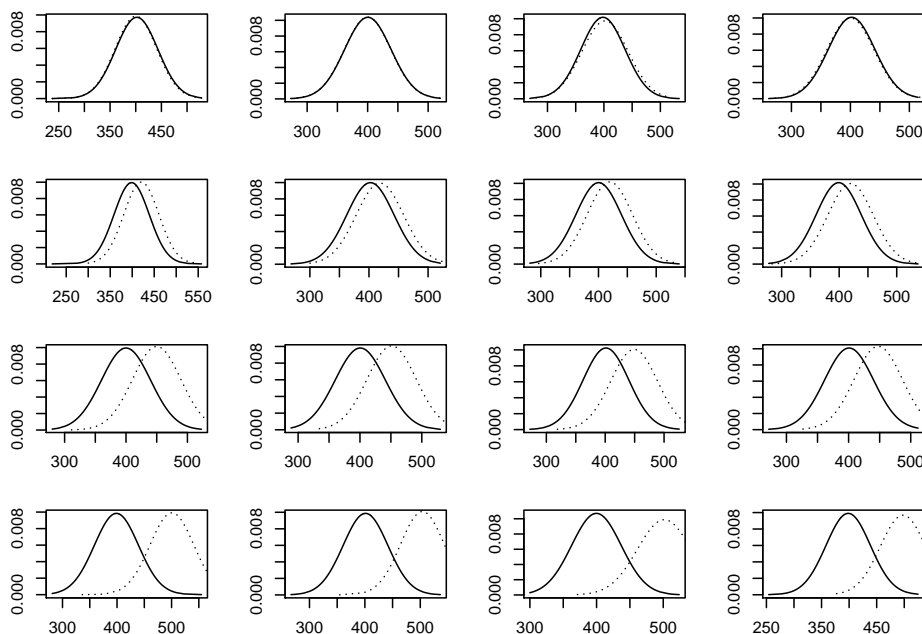


Figura 5.4: Distribución de los datos en un supersujeto. Efectos sobre la media. Sin outliers

Fijando el valor del efecto sobre  $\mu$ , al comparar los casos  $\frac{1}{TR}$  y  $\frac{1}{TR^2}$  se observa más uniformidad en la diferencia entre los casos sin outliers y con un 10 % de outliers. En concreto, con efectos de  $100$  ms., la diferencia entre los casos sin outliers y con un 10 % de outliers en la izquierda de la gráfica se reduce considerablemente. En concreto, la diferencia entre los casos sin outliers y con un 10 % de outliers se reduce notablemente con efectos grandes al pasar de la transformación  $\frac{1}{TR}$  a  $\frac{1}{TR^2}$ .

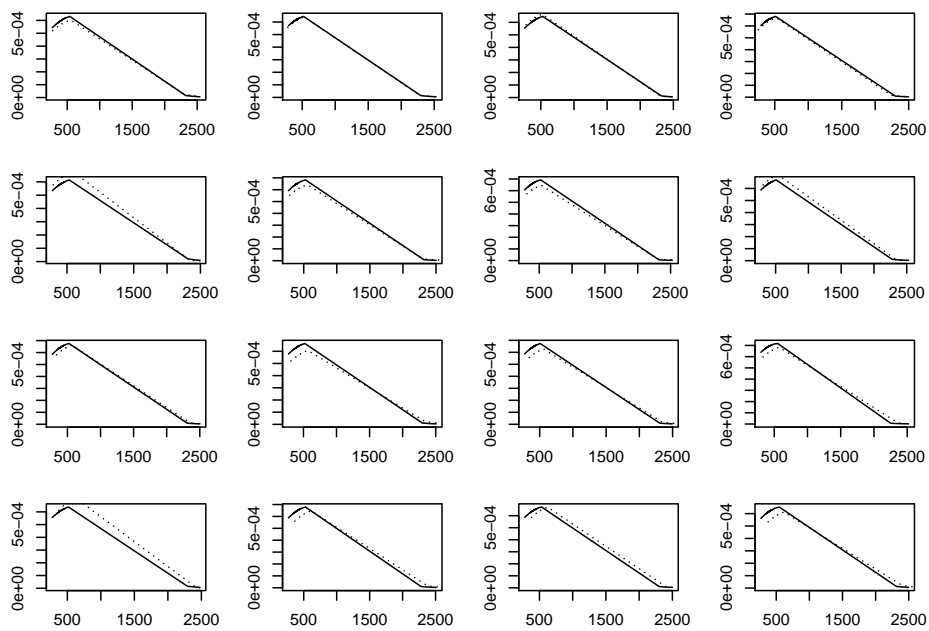


Figura 5.5: Distribución de los datos en un supersujeto. Efectos sobre la media.  
10 % outliers

En la transformación  $\frac{1}{TR}$  la diferencia entre los casos sin outliers y con un 10 % de outliers oscila. Sin embargo, esta diferencia parece mantener mayor estabilidad en la transformación  $\frac{1}{TR^2}$ .

### 5.4.2 Efectos sobre la cola

De forma similar a como se estudió gráficamente la distribución de los datos al aplicar efectos sobre la media, los efectos sobre la cola  $\tau_{ef}$  toman posibles valores 0, 20, 50 y 100 ms. (cada fila) y las posibles transformaciones se muestran en cada columna.

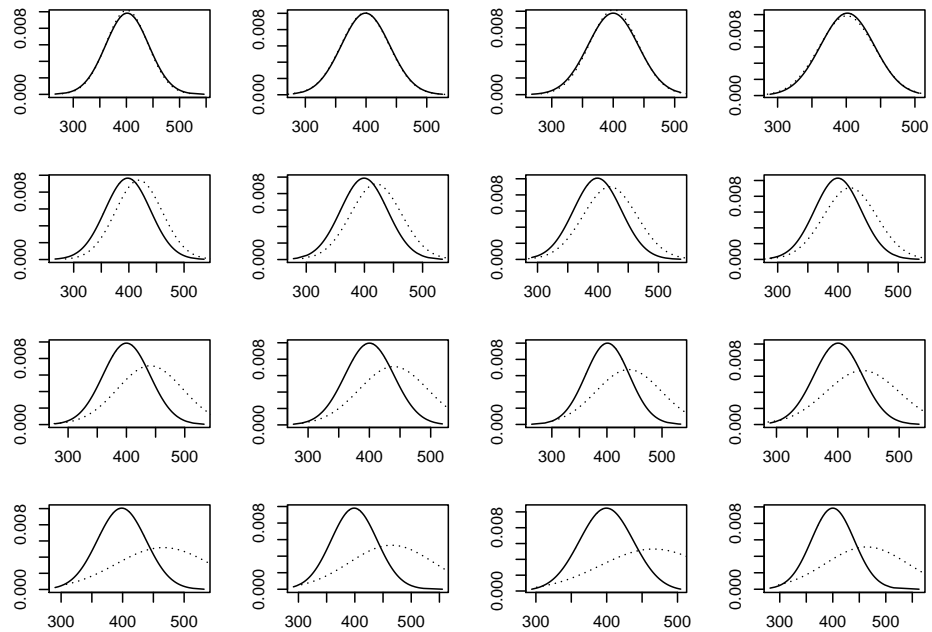


Figura 5.6: Distribución de los datos en un supersujeto. Efectos sobre la cola. Sin outliers

Sin outliers, los efectos sobre la cola no ofrecen diferencias significativas entre los gráficos sin y con efecto para los diferentes valores de los mismos.

Al analizar la situación con un 10 % de outliers sí se observa diferencia entre los gráficos. La presencia de un 10 % de outliers hace que la pendiente de la distribución sea uniforme en toda la gráfica salvo en los valores más extremos. Con efectos sobre la cola inferiores a 50 ms. la diferencia entre los casos sin y

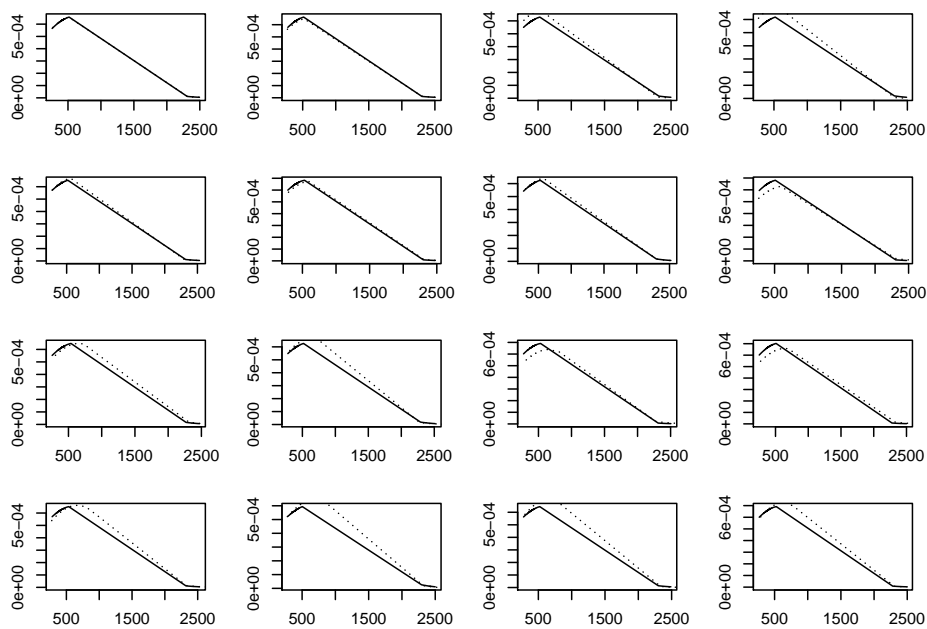


Figura 5.7: Distribución de los datos en un supersujeto. Efectos sobre la cola 10 % outliers

con efecto no se aprecia significativamente. Sin embargo, en los casos de efecto sobre la cola de *50* y *100 ms.* sí puede observarse mejor adaptación en el caso  $\frac{1}{TR}$  frente a  $\frac{1}{TR^2}$ .

## 5.5 Análisis de dispersión de los datos

Se analiza en este apartado la dispersión comparativa de los datos sometidos a las diferentes transformaciones mediante el empleo del *Coficiente de Variación de Pearson*. Cada gráfico muestra los casos sin outliers (gris) y con un 10 % de outliers (negro) sometidos a las diferentes transformaciones posibles. Los datos se distribuyen en base a la distribución *ExGauss* con los parámetros habituales y aplicando efectos sobre  $\mu$  y  $\tau$  de valores *0*, *20*, *50* y *100 ms.* en los respectivos bloques de gráficos.

Puede verse a continuación el código fuente empleado para el cálculo del Coeficiente de Variación de Pearson a partir de datos generados aleatoriamente.

```
outliers_glme_dispersion<-function(mu=400,sigma=40,tau=200,mu.ef=30,
                                   sigma.ef=0,tau.ef=0,p.outlier=0.1,nsujetos=32,nitems
{
  # Generación de una muestra aleatoria

  A<-matrix(nrow=1,ncol=4);

  medias_base<-rnorm(n=nitems,mean=mu,sd=sigma)+rexp(n=1,rate=1/tau);

  for(i in 1:nsujetos)
  {
    A1<-matrix(nrow=nitems*2,ncol=4);
    muestra1<-rep(x=0,times=nitems);
    muestra2<-rep(x=0,times=nitems);

    for(j in 1:nitems)
    {
      # Condición 1
      muestra1[j]<-rnorm(n=1,mean=medias_base[j],sd=sigma)+
      rexp(n=1,rate=1/tau);
```

```

# Condición 2
muestra2[j]<-rnorm(n=1,mean=medias_base[j]+mu.ef,sd=sigma+sigma.ef)+
  rexp(n=1,rate=(1/tau)+(1/tau.ef));
}

aleatorio<-runif(n=nitems,min=0,max=1);
muestra1[which(aleatorio<p.outlier)]<-
muestra1[which(aleatorio<p.outlier)]+2000;

aleatorio<-runif(n=nitems,min=0,max=1);
muestra2[which(aleatorio<p.outlier)]<-
muestra2[which(aleatorio<p.outlier)]+2000;

if(metodo=="log")      # Transformación logaritmo
{
  muestra1<-log(x=muestra1);
  muestra2<-log(x=muestra2);
}

if(metodo=="inv")      # Transformación inversa
{
  muestra1<-1/muestra1;
  muestra2<-1/muestra2;
}

if(metodo=="inv2")     # Transformación inversa del cuadrado
{
  muestra1<-1/(muestra1^2);
  muestra2<-1/(muestra2^2);
}

for (k in 1:nitems)
{
  A1[k,]<-c(i,k,1,muestra1[k]);
  A1[k+nitems,]<-c(i,k,2,muestra2[k]);
}

```

```

    A<-rbind(A,A1);
  }

  # La primera fila está vacía por ser parte de la creación de A y la elimino
  A<-A[-1,];

  # Organizamos los datos en un Data Frame para poder tratarlos mediante la
# función lmer de la librería lme4

  a<-A[,4];

  r<-sqrt(var(a))/mean(a);

  return(r);
}

```

### 5.5.1 Efectos sobre la media

Cada gráfico, de izquierda a derecha representa que se aplica un efecto de  $\theta$ , 20, 50 y 100 ms. sobre  $\mu$ .

En todos los valores estudiados de efecto sobre la media, tal y como era previsible, es mayor la dispersión para cada transformación en el caso sin outliers (gris) que en el caso con un 10 % de outliers (negro). Al aumentar el efecto, centrándonos en la transformación  $\frac{1}{T R^2}$  los valores de la dispersión aumentan en el caso sin outliers pero se mantienen estables con un 10 % de outliers. La consecuencia es que la diferencia entre ambas casuísticas se reduce. Con efectos de hasta 50 ms. la diferencia de dispersión entre los casos sin outliers y con un 10 % de outliers no es notable.

### 5.5.2 Efectos sobre la cola

Cada gráfico, de izquierda a derecha representa que se aplica un efecto de  $\theta$ , 20, 50 y 100 ms. sobre  $\tau$ .

Con un 10 % de outliers la dispersión se mantiene estable al aumentar el



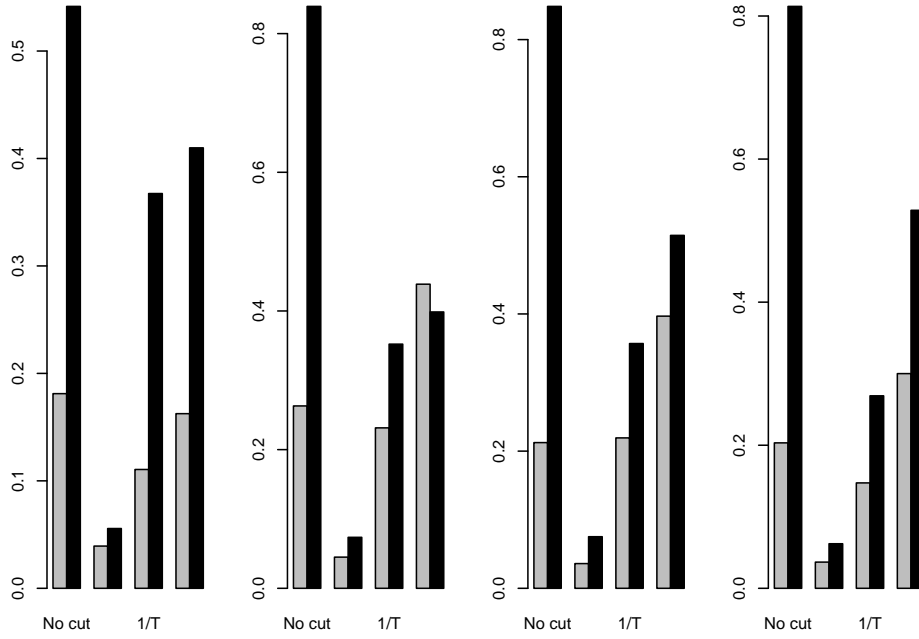


Figura 5.8: Dispersión de los datos en un supersujeto. Efectos sobre la media

efecto sobre  $\tau$ . La menor diferencia entre los casos sin outliers y con un 10 % de outliers se observa para efectos pequeños (20 ms). Sin embargo, aunque al aumentar los efectos aumentan las diferencias entre ambas casuísticas para la dispersión, los valores de la dispersión se reducen. Tal y como se termina de introducir, nótese el caso con efectos mínimos sobre  $\tau$  (20 ms.) como aquel en que se observa mayores valores de la dispersión para la transformación  $\frac{1}{TR^2}$  pero menor diferencia entre ambos casos.

## 5.6 Conclusiones

Si estudiamos la dispersión de los datos frente a las diferentes transformaciones y con diferentes efectos:

- Con efectos sobre  $\mu$  hasta 50 ms. la dispersión se reduce con cada transformación
- Con efectos sobre  $\tau$  hasta 50 ms. la dispersión se reduce con cada transformación. Aumentando el efecto hasta los 100 ms., aunque los valores de

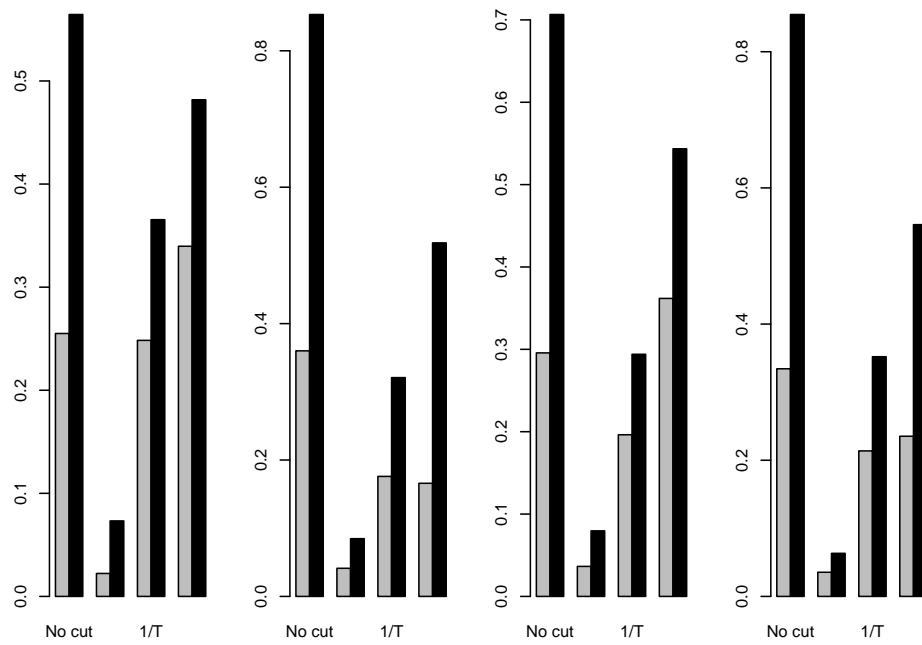


Figura 5.9: Dispersión de los datos en un supersujeto. Efectos sobre la cola

dispersión aumentan, se reduce notablemente la diferencia entre los valores de dispersión para los casos sin outliers y con un 10 % de outliers.

## Capítulo 6

# Tratamiento de Outliers en Tiempos de Respuesta de reconocimiento de palabras mediante transformaciones de datos y análisis de resultados mediante Modelos Lineales

### 6.1 Introducción

En el trabajo de Ratcliff (1993) se completó un análisis de diferentes técnicas de tratamiento de Outliers para  $TR$  en reconocimiento de palabras con dos condiciones,  $n_S$  sujetos y  $n_I$  items por sujeto. Se evaluaron, entre otras, las transformaciones *Nocut* (sin transformación),  $\log TR$  y  $\frac{1}{TR}$ . y los resultados fueron analizados mediante la técnica clásica *t-test*. Se concluyó que  $\frac{1}{TR}$  ofrecía

la mayor potencia. El objetivo del presente desarrollo es hacer el mismo tipo de análisis sobre el mismo tipo de condiciones incorporando la transformación  $\frac{1}{TR^2}$ .

### 6.1.1 Resumen conceptual del Modelo Lineal

Los conceptos básicos de un Modelo Lineal de predicción pueden repasarse en el trabajo de Bodo (2013b).

Si estudiamos dos características o variables aleatorias ( $x$  e  $y$ ) en el mismo grupo de sujetos obtenemos una relación

$$y \sim x$$

Podemos decir que “ $y$  depende de  $x$ ”, o en términos matemáticos clásicos  $x$  es la variable independiente e  $y$  es variable dependiente. Una relación de este tipo sólo es válida en una situación ideal en que la variable  $y$  se considera exclusivamente dependiente de  $x$ . Podríamos incorporar al modelo o ecuación anterior un componente aleatorio o probabilístico, que expresaremos  $\epsilon$  que explique la parte no fijada del segundo miembro de la ecuación, de forma que

$$y \sim x + \epsilon$$

Podríamos añadir al modelo lineal anterior otras variables independientes  $x_1, x_2, \dots$  que determinaran la variable  $y$ , además del componente aleatorio  $\epsilon$ , de forma que

$$y \sim x_1 + x_2 + \dots + \epsilon$$

Todos los casos anteriores son modelos lineales, es decir,  $y$  queda determinado mediante una combinación lineal de  $x_1, x_2, \dots$

Veamos ahora las condiciones que determinan la linealidad de un modelo.

1. **Linealidad.** Es suficiente representar el gráfico de predicciones frente al de los residuales para observar si los residuos siguen un patrón lineal o si los mismos residuos siguen un patrón no linealidad.
2. **Ausencia de colinealidad.** Cuando dos efectos fijos o predictores están relacionados entre sí diremos que existe colinealidad entre ambos. Si se produce esta situación, la interpretación de los resultados se complica

dentro del marco base de un modelo predictivo lineal. Debe plantearse el experimento inicial evitando poder incurrir en situaciones de colinealidad.

3. **Homocedasticidad o ausencia de heterocedasticidad.** Un modelo presenta homocedasticidad cuando la varianza del error para los diferentes efectos fijos es constante a lo largo de las observaciones. Tanto la colinealidad (ausencia de) como la homocedasticidad pueden tratarse de controlarse aplicando diferentes transformaciones sobre los datos, tema de estudio en concreto del presente trabajo.
4. **Normalidad de los residuos.** La normalidad de los residuos es una cuestión cuya importancia para determinar la linealidad del modelo es controvertida entre los diferentes modelos. Particularmente, el texto de Gellman & Hill (2007) no recomienda dicho estudio de normalidad de residuos. Para su estudio dispondríamos de dos opciones: *a)* histograma de residuos (`hist`), y *b)* Q-Q plot que representa la regresión lineal de los propios residuos obtenidos del modelo lineal de ajuste previo (`qqnorm`).
5. **Ausencia de puntos de influencia.** Puede utilizarse en R la función `DFBeta` para evaluar el peso específico de cada efecto dentro del modelo lineal, es decir, la variación del correspondiente coeficiente si se descartara el uso de esta variable en el modelo final.
6. **Independencia.** Se ha demostrado en diferentes trabajos y ámbitos de estudio que no es infrecuente incumplir la independencia estadística de las lecturas. Sirvan como ejemplos los trabajos de Freeberg & Lucas (2009) en Psicología, Lazic (2010) en Neurociencia o Winter (2011) en Neurolingüística.

### 6.1.2 Planteamiento del problema y Modelo Lineal

Consideremos 2 condiciones,  $n_S$  sujetos y  $n_I$  items. Bajo la primera condición pueden por tanto considerarse  $n_S$  muestras de  $n_I$  items independientes e idénticamente distribuidas en base a una distribución *ExGauss* de parámetros  $\mu$ ,  $\sigma$  y  $\tau$ .

Si los  $n_S$  sujetos se someten a los mismos  $n_I$  items debemos tener en cuenta la variabilidad entre items dentro de un mismo sujeto y también la variabilidad entre sujetos para el mismo item. Particularmente, partiendo de *ExGauss*( $\mu, \sigma, \tau$ ), para cada sujeto  $\mu \sim \text{ExGauss}(\mu, \sigma, \tau)$ ,  $\sigma \sim N(\sigma, \frac{\sigma}{2})$  y  $\tau \sim N(\tau, \frac{\tau}{2})$ .

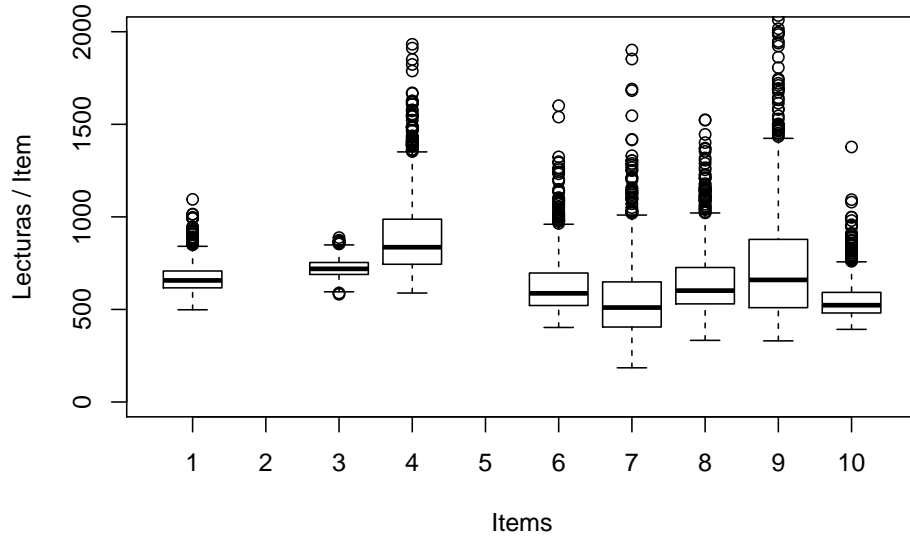


Figura 6.1: Distribución ExGauss de 10 items

Las  $n_S$  muestras del segundo sujeto emplean efectos sobre los tres parámetros de la distribución *ExGauss* original dados por  $\mu_{ef}$ ,  $\sigma_{ef}$  y  $\tau_{ef}$ .

Por tanto, podemos considerar cada *TR* analizado como perteneciente a un posible sujeto, item y condición. El análisis de datos, como en Ratcliff (1993), se hará mediante un modelo lineal.

El análisis computacional se hará empleando el software estadístico **R-Project** (2006), embebido dentro de su *FrontEnd R-studio* (2015), que se ha empleado para la redacción del presente documento, conjuntamente con el sistema de procesamiento de textos TeX (1984). Las múltiples simulaciones estadísticas que han permitido generar las muestras respectivas y analizar los correspondientes resultados se han ejecutado en el entorno del *Supercomputador Lluís Vives* de la *Universitat de València*, cuya información detallada puede consultarse en la web del *Centro de Cálculo* de la *Universitat de València* («WIKI del Servicio de Informática SiUV», 2008).

Los datos se organizan en una estructura *Data Frame* de 4 columnas del tipo

#|Subj|Item|Cond.|TR|

tal y como se muestra en Baayen, Davidson & Bates (2008). La primera columna indica el sujeto (valores de 1 a  $n_S$ ), la segunda el item (valores de 1 a

$n_I$ ), la columna *SOA* indica la condición (valores 1 y 2) y finalmente la columna *TR* con los Tiempos de Respuesta o las correspondientes transformaciones de los mismos. Por tanto, estamos hablando de una tabla de datos con  $n_S \times 2 \times n_I$  entradas.

Las cuatro transformaciones posibles de un *TR* son:  $t$  (sin transformación o No cut),  $\log(t)$  (logaritmo),  $\frac{1}{t}$  (inversa) y  $\frac{1}{t^2}$  (cuadrado o potencia 2 de la inversa). Estas funciones transformación son positivas y crecientes si  $t > 0$  y además

$$\frac{1}{t^2} < \frac{1}{t} < \log(t) < t, \forall t \in ]1, +\infty[$$

Matemáticamente, consideraremos los *TR* generados en base a una distribución ExGauss y expresados en *ms*. Sean  $\mu$ ,  $\sigma$  y  $\tau$  los parámetros que definen la media, desviación típica y cola de la distribución ExGauss  $T$  generada como suma de las distribuciones  $Ex(\tau)$  y  $N(\mu, \sigma)$ . Conocidas las distribuciones de ambas, la función de densidad de la *ExGauss* resultante es de la forma

$$f(t|\mu, \sigma, \tau) = \tau \hat{u} e^{-\tau \hat{u} t} + \frac{1}{\sigma \hat{u} \sqrt{2\hat{u}\pi}} \hat{u} e^{-\frac{1}{2}\hat{u}(\frac{t-\mu}{\sigma})^2}, t > 0$$

. Por el mismo motivo, podemos obtener la esperanza y varianza de la distribución ExGauss de los TR.

$$E(T) = \frac{1}{\tau} + \mu, Var(T) = \frac{1}{\tau^2} + \sigma^2, t > 0$$

Particularmente, la distribución *ExGauss* básica de los experimentos que siguen tiene por parámetros  $\mu = 400$ ,  $\sigma = 40$  y  $\tau = 200$ .

El modelo lineal empleado para el análisis de resultados será, con la notación R del tipo

$$RT \sim SOA + Item + Subj$$

y el correspondiente análisis de resultados se hará mediante un ANOVA o **t-test** tradicional tal y como se hizo en el trabajo original de Ratcliff (1993).

## 6.2 Análisis de potencia

En cada sección de este apartado se muestra un gráfico múltiple en que se indica de arriba a abajo el análisis para 10, 20, 30 y 40 sujetos. De izquierda

a derecha 10, 20, 30 y 40 items. En cada gráfico se muestra el resultado para cada tipo de transformación sin outliers y con un 10 % de outliers.

### 6.2.1 Análisis error tipo I

Sin efecto sobre ninguna de las variables. Mismas condiciones que en Ratcliff (1993).

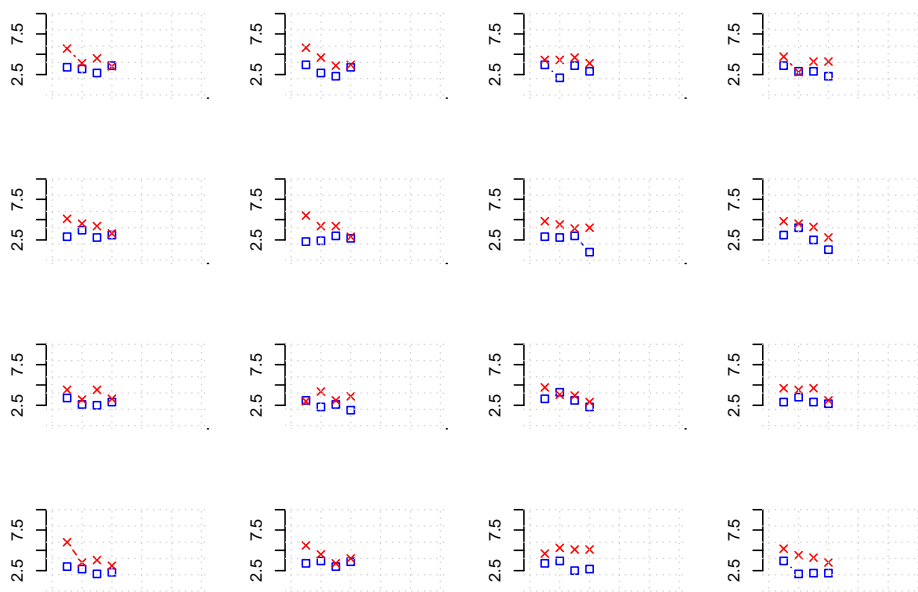


Figura 6.2: Análisis de falsos positivos

No aplicando efecto sobre ninguno de los parámetros de la distribución *Ex-Gauss* se observa que la proporción de falsos positivos se mantiene alrededor del 5 %. En concreto, con número de sujetos y/o items creciente, se observa sensible disminución al emplear la transformación  $\frac{1}{t^2}$ , así como una reducción de la diferencia entre los casos sin outliers y con un 10 % de outliers.

### 6.2.2 Efectos sobre la media

Al aumentar número de sujetos y/o items, con un efecto de 20 ms. aumentan los valores en la gráfica a la par que disminuye la diferencia entre los casos sin outliers y con un 10 % de outliers.



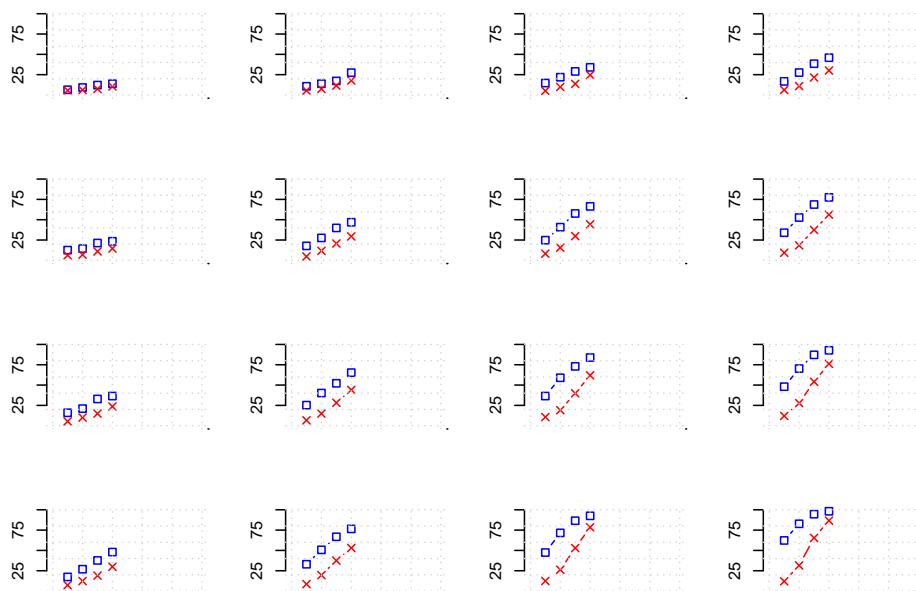


Figura 6.3: Efecto de 20 ms. sobre la media.

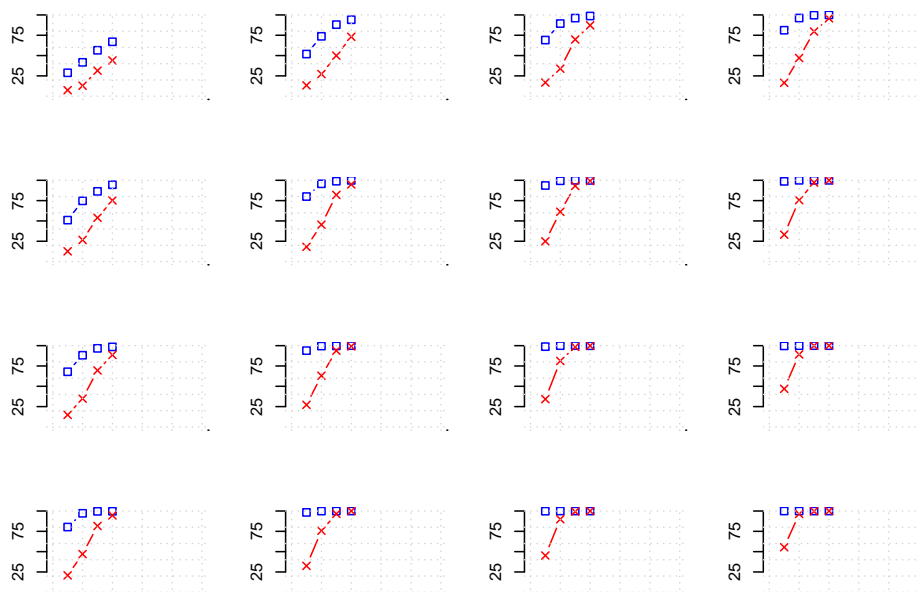


Figura 6.4: Efecto de 50 ms. sobre la media

Con efectos de  $50\text{ ms.}$  no se observa diferencia entre las transformaciones inversa y cuadrado de la inversa a para un número de sujetos superior a  $20$ .

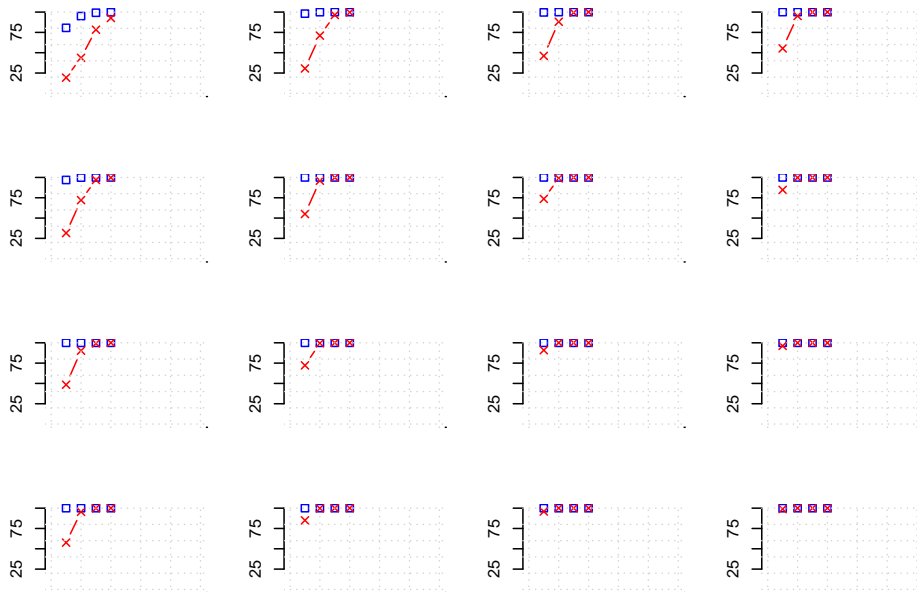


Figura 6.5: Efecto de  $100\text{ ms.}$  sobre la media

La situación expresada en el caso anterior, con efectos de  $50\text{ ms.}$  sobre la media, se ve incrementada al ampliar el efecto sobre la media hasta  $100\text{ ms.}$ .

### 6.2.3 Efectos sobre la cola

Un efecto pequeño sobre la cola de  $20\text{ ms.}$  hace casi imperceptible la diferencia entre los casos sin outliers y con un  $10\%$  de outliers con número de sujetos y/o items pequeño. Al aumentar este número de sujetos y/o items, aunque la diferencia entre los casos sin outliers y con un  $10\%$  de outliers se incrementa en las primeras tres posibles transformaciones, se mantiene estable la diferencia entre los dos casos con la transformación  $\frac{1}{t^2}$ .

Al aumentar el efecto sobre la cola hasta los  $50\text{ ms.}$  la difefencia entre los casos sin outliers y con un  $10\%$  de outliers disminuye.

Con efecto grande, más de  $100\text{ ms.}$  deja de apreciarse diferencia, tanto a nivel de items y/o sujetos, como entre transformaciones, siempre comparando

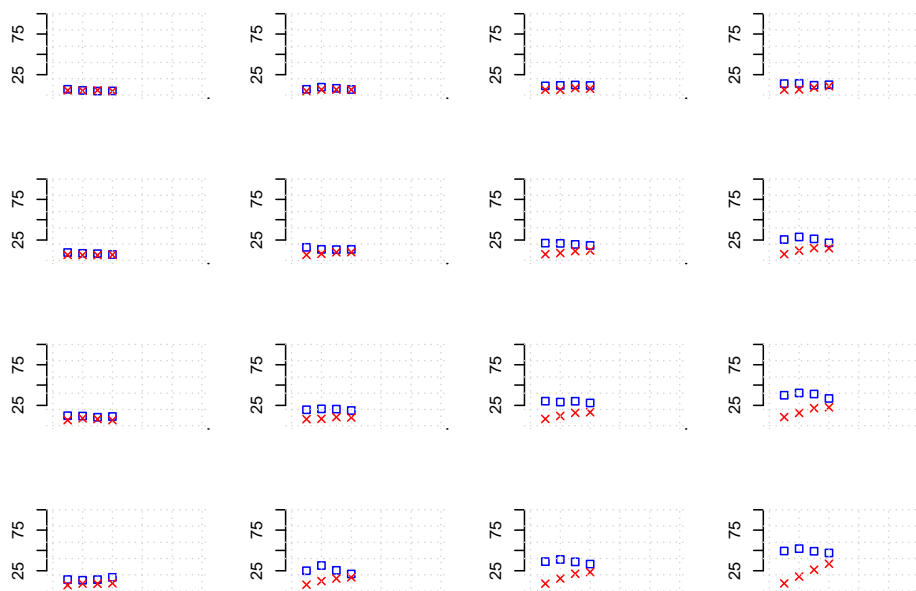


Figura 6.6: Efecto de 20 ms. sobre la cola

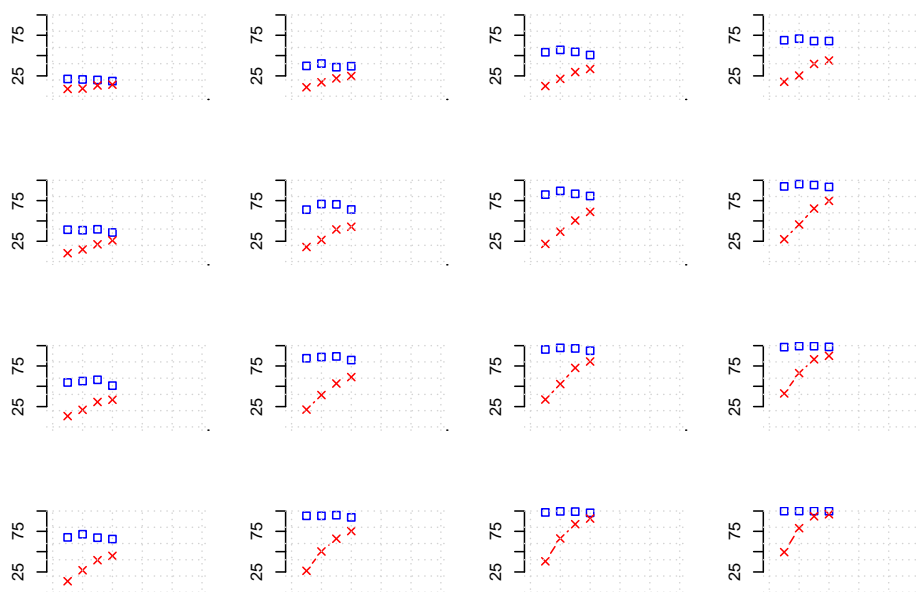


Figura 6.7: Efecto de 50 ms. sobre la cola

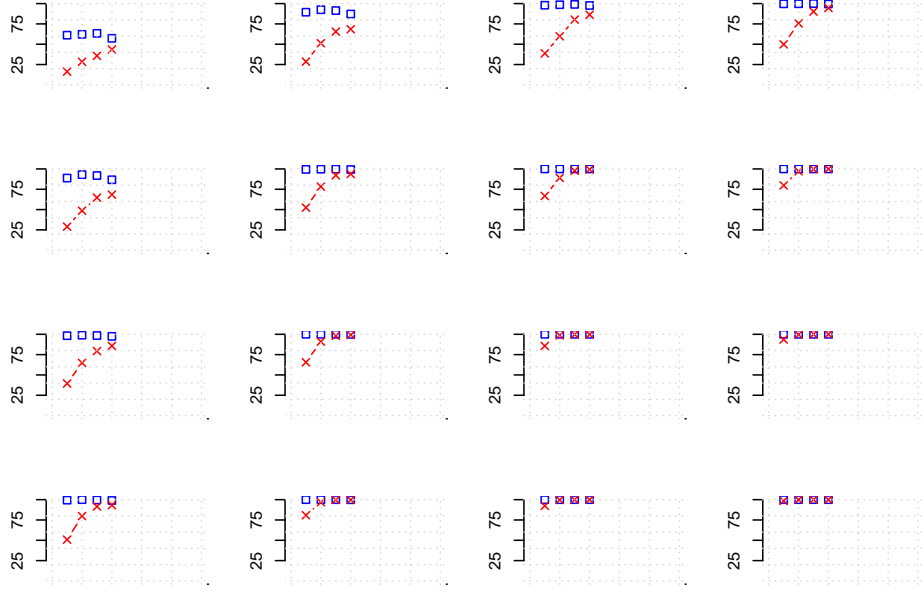


Figura 6.8: Efecto de 100 ms. sobre la cola

los casos sin outliers y con un  $10\%$  de outliers.

### 6.3 Análisis de residuos

La pretensión de esta sección es el análisis de los residuos generados con cada tipo de transformación al aplicar el modelo lineal.

Recordemos, según se indicó en el planteamiento del modelo, que partiremos de 32 sujetos representados por 7 items por sujeto y 2 condiciones, acorde a las condiciones de Ratcliff (1993). Es decir, 32x2 simulaciones *ExGauss* de muestras de tamaño 7. Esto puede estructurarse mentalmente como 2 matrices (una por condición) de 32 filas (sujetos) y 7 columnas (items) por condición. La distribución *ExGauss* básica se basa en parámetros  $\mu = 400$ ,  $\sigma = 40$  y  $\tau = 200$ .

Se aplicarán diferentes efectos sobre los parámetros  $\mu$ ,  $\sigma$  y  $\tau$  de la distribución *ExGauss* original.

Además, en las diferentes muestras dispondremos de un máximo de  $10\%$  de outliers.

Para cada variación de los parámetros se mostrarán y analizarán tres clases de gráficos, que pretenden analizar cada uno un aspecto diferente. Los tres tipos de gráficos (múltiples) se organizan en una presentación de  $4 \times 3$  subgráficos, cada fila analiza un tipo de transformación de los ya planteados y las columnas analizan los casos sin outliers y con un 10 % de outliers.

Los tres tipos de gráficos son:

- *QQ-Plot*. Comparación de los residuos con la recta de regresión lineal. Una única repetición del experimento.
- *Normalidad de residuos*. Histograma de residuos con una única repetición del experimento. La normalidad se analiza mediante el *Test de Pearson*  $\chi^2$ . El gráfico resultante será de color rojo si la distribución no es normal y azul si el resultado del test anterior es positivo. El uso del *Test de Pearson* se ha hecho mediante el uso de la librería de R **Nortest** (2015).
- *Histograma de residuos*. Como los dos primeros tipos de gráficos se basan en una única repetición del experimento correspondiente es necesario hacer un análisis estadístico basado en múltiples repeticiones. Tras 1000 repeticiones del experimento se muestra el histograma de las normas euclídeas de los vectores de residuos generados en cada repetición.

A continuación el código fuente de la generación de los gráficos recién señalados en las líneas previas.

```
outliers_lme_residuals_qq<-function(mu=400,sigma=40,tau=200,mu.ef=30,sigma.ef=0,tau.ef=0,p.outliers)
{
  # Generación de una muestra aleatoria

  A<-matrix(nrow=1,ncol=4);

  medias_base<-rnorm(n=nitems,mean=mu,sd=sigma)+rexp(n=1,rate=1/tau);

  for(i in 1:nsujetos)
  {
    A1<-matrix(nrow=nitems*2,ncol=4);
    muestra1<-rep(x=0,times=nitems);
    muestra2<-rep(x=0,times=nitems);

    for(j in 1:nitems)
```

```

{
  # Condición 1
  muestra1[j]<-rnorm(n=1,mean=medias_base[j],sd=sigma)+
  rexp(n=1,rate=1/tau);

  # Condición 2
  muestra2[j]<-rnorm(n=1,mean=medias_base[j]+mu.ef,sd=sigma+sigma.ef)+
  rexp(n=1,rate=(1/tau)+(1/tau.ef));
}

aleatorio<-runif(n=nitems,min=0,max=1);
muestra1[which(aleatorio<p.outlier)]<-muestra1[which(aleatorio<p.outlier)]
+2000;

aleatorio<-runif(n=nitems,min=0,max=1);
muestra2[which(aleatorio<p.outlier)]<-muestra2[which(aleatorio<p.outlier)]
+2000;

if(metodo=="log")      # Transformación logarítmica
{
  muestra1<-log(x=muestra1);
  muestra2<-log(x=muestra2);
}

if(metodo=="inv")      # Transformación inversa
{
  muestra1<-1/muestra1;
  muestra2<-1/muestra2;
}

if(metodo=="inv2")     # Transformación inversa del cuadrado
{
  muestra1<-1/(muestra1^2);
  muestra2<-1/(muestra2^2);
}

```

```

for (k in 1:nitems)
{
  A1[k,]<-c(i,k,1,muestra1[k]);
  A1[k+nitems,]<-c(i,k,2,muestra2[k]);
}

A<-rbind(A,A1);
}

# La primera fila está vacía por ser parte de la creación de A y la elimino
A<-A[-1,];

# Organizamos los datos en un Data Frame para poder tratarlos mediante
# la función lmer de la librería lme4

datos<-data.frame(Subj=A[,1],Item=A[,2],SOA=A[,3],RT=A[,4]);

lm<-lm(RT~SOA+Item+Subj,data=datos,REML=F);

Res<-summary(lm)$residuals; # residuos
qqnorm(Res,main="");
qqline(Res);
}

outliers_lme_residuals_qqhist<-function(mu=400,sigma=40,tau=200,mu.ef=30,sigma.ef=0,tau.ef=0,p.out)
{
  v_histograma<-rep(x=0,times=nrep);

  for(nrepeticion in 1:nrep)
  {
    # Generación de una muestra aleatoria

    A<-matrix(nrow=1,ncol=4);

    medias_base<-rnorm(n=nitems,mean=mu,sd=sigma)+rexp(n=1,rate=1/tau);

    for(i in 1:nsujetos)

```

```

{
  A1<-matrix(nrow=nitems*2,ncol=4);
  muestra1<-rep(x=0,times=nitems);
  muestra2<-rep(x=0,times=nitems);

  for(j in 1:nitems)
  {
    # Condición 1
    muestra1[j]<-rnorm(n=1,mean=medias_base[j],sd=sigma)+
    rexp(n=1,rate=1/tau);

    # Condición 2
    muestra2[j]<-rnorm(n=1,mean=medias_base[j]+mu.ef,
    sd=sigma+sigma.ef)+rexp(n=1,rate=(1/tau)+(1/tau.ef));
  }

  aleatorio<-runif(n=nitems,min=0,max=1);
  muestra1[which(aleatorio<p.outlier)]<-
  muestra1[which(aleatorio<p.outlier)]+2000;

  aleatorio<-runif(n=nitems,min=0,max=1);
  muestra2[which(aleatorio<p.outlier)]<-
  muestra2[which(aleatorio<p.outlier)]+2000;

  if(metodo=="log")      # Transformación logarítmica
  {
    muestra1<-log(x=muestra1);
    muestra2<-log(x=muestra2);
  }

  if(metodo=="inv")      # Transformación inversa
  {
    muestra1<-1/muestra1;
    muestra2<-1/muestra2;
  }
}

```



```

if(metodo=="inv2")      # Transformación inversa del cuadrado
{
  muestra1<-1/(muestra1^2);
  muestra2<-1/(muestra2^2);
}

for (k in 1:nitems)
{
  A1[k,]<-c(i,k,1,muestra1[k]);
  A1[k+nitems,]<-c(i,k,2,muestra2[k]);
}
A<-rbind(A,A1);
}

# La primera fila está vacía por ser parte de la creación de A y la elimino
A<-A[-1,];

# Organizamos los datos en un Data Frame para poder tratarlos mediante un
# modelo lineal

datos<-data.frame(Subj=A[,1],Item=A[,2],SOA=A[,3],RT=A[,4]);

lm<-lm(RT~SOA+Item+Subj,data=datos,REML=F);

Res<-summary(lm)$residuals; # residuos
x <- c(qnorm(.25), qnorm(.75));
y <- quantile(Res, c(.25, .75));
pdte <- diff(y)/diff(x);
int <- y[1] - pdte * x[1];
diferencia=rep(x=0,times=nsujetos*nitems*2);
for(k in 1:nsujetos*nitems*2)
{
  # diferencia entre los residuos y la recta QQ
  diferencia[k]<-Res[k]-(int+pdte*Res[k]);
}
v_histograma[nrepeticion]<-sqrt(sum((Res-diferencia)^2));

```

```

    }

    return(v_histograma);
}

outliers_lme_residuals_hist<-function(mu=400,sigma=40,tau=200,mu.ef=30,sigma.ef=0,tau.ef=200)
{
  # Generación de una muestra aleatoria

  A<-matrix(nrow=1,ncol=4);

  medias_base<-rnorm(n=nitems,mean=mu,sd=sigma)+rexp(n=1,rate=1/tau);

  for(i in 1:nsujetos)
  {
    A1<-matrix(nrow=nitems*2,ncol=4);
    muestra1<-rep(x=0,times=nitems);
    muestra2<-rep(x=0,times=nitems);

    for(j in 1:nitems)
    {
      # Condición 1
      muestra1[j]<-rnorm(n=1,mean=medias_base[j],
        sd=sigma)+rexp(n=1,rate=1/tau);

      # Condición 2
      muestra2[j]<-rnorm(n=1,mean=medias_base[j]+mu.ef,sd=sigma+sigma.ef)+
        rexp(n=1,rate=(1/tau)+(1/tau.ef));
    }

    aleatorio<-runif(n=nitems,min=0,max=1);
    muestra1[which(aleatorio<p.outlier)]<-muestra1[which(aleatorio<p.outlier)]
    +2000;

    aleatorio<-runif(n=nitems,min=0,max=1);
    muestra2[which(aleatorio<p.outlier)]<-muestra2[which(aleatorio<p.outlier)]
    +2000;
  }
}

```

```

if(metodo=="log")      # Transformación logaritmo
{
  muestra1<-log(x=muestra1);
  muestra2<-log(x=muestra2);
}

if(metodo=="inv")      # Transformación inversa
{
  muestra1<-1/muestra1;
  muestra2<-1/muestra2;
}

if(metodo=="inv2")     # Transformación inversa del cuadrado
{
  muestra1<-1/(muestra1^2);
  muestra2<-1/(muestra2^2);
}

for (k in 1:nitems)
{
  A1[k,]<-c(i,k,1,muestra1[k]);
  A1[k+nitems,]<-c(i,k,2,muestra2[k]);
}

A<-rbind(A,A1);
}

# La primera fila está vacía por ser parte de la creación de A y la elimino
A<-A[-1,];

# Organizamos los datos en un Data Frame para poder tratarlos mediante
# la función lmer de la librería lme4

datos<-data.frame(Subj=A[,1],Item=A[,2],SOA=A[,3],RT=A[,4]);

lm<-lm(RT~SOA+Item+Subj,data=datos,REML=F);

```

```

Res<-summary(lm)$residuals; # residuos
# hist(Res,main="Histogram of residuals",xlab="Residuals");
pvalor<-pearson.test(Res)$p.value;

if (pvalor>=0.5)
{
  hist(Res,xlab="",ylab="",main="",border="blue",breaks=20);
}
if (pvalor<0.5)
{
  hist(Res,xlab="",ylab="",main="",border="red",breaks=20);
}
media<-mean(Res);
desviacion<-sd(Res);
abline(v=media,lty="dashed",col="green");
abline(v=media-desviacion,lty="dashed",col="yellow");
abline(v=media+desviacion,lty="dashed",col="yellow");
}

```

### 6.3.1 Sin efecto sobre ningún parámetro ni variabilidad entre sujetos

## Loading required package: Matrix

Sin efecto sobre ningún parámetro ni variabilidad entre sujetos, se observa que cada transformación observa mejor adaptación de los residuos a la recta Cuantil-Cuantil que la transformación anterior. Nótese, concretamente en el caso con un 10 % de outliers, la mejor adaptación del caso  $\frac{1}{TR^2}$  frente al caso  $\frac{1}{TR}$ .

En el caso con un 10 % de outliers, al pasar de la transformación  $\frac{1}{TR}$  a  $\frac{1}{TR^2}$  hay una notable mejora de la simetría respecto a la media.

### 6.3.2 Efecto sobre la media y sin variabilidad entre sujetos

Con un efecto pequeño sobre la media, la adaptación de los residuos a la recta Cuantil-Cuantil es notablemente mejor con la última transformación aplicada,

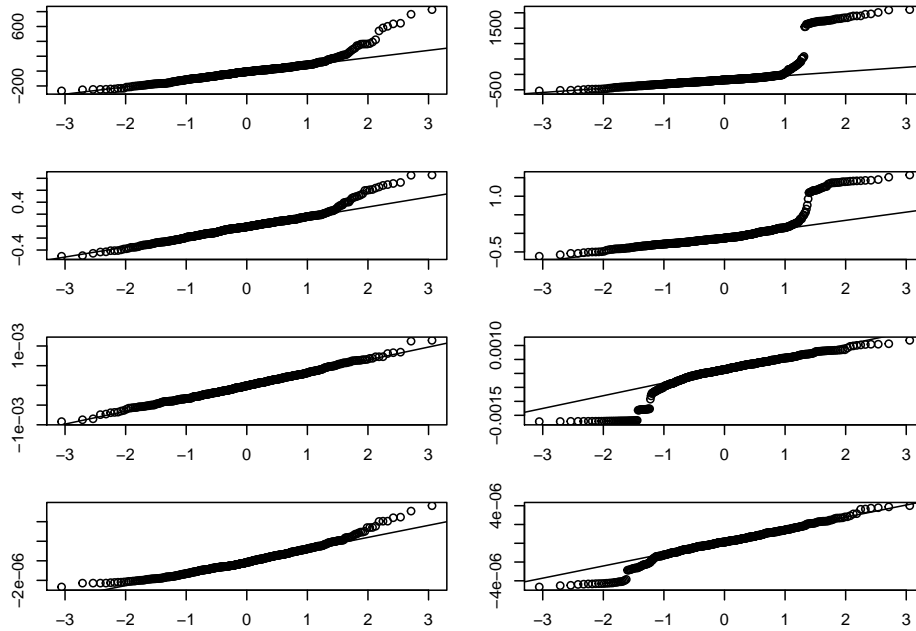


Figura 6.9: Sin efectos ni variabilidad intersujetos. QQ-Norm.

notablemente en el caso con un 10 % de outliers.

Un efecto de 20 ms. sobre la media sólo afecta aparentemente en una reducción de la media, analizando la suma de residuos.

En un único experimento, la transformación  $\frac{1}{TR^2}$  parece disminuir el número de residuos alejados de la distribución normal, en concreto los residuos del caso sin outliers. Adaptación a la recta Cuantil-Cuantil no tan notable en el caso con un 10 % de outliers.

Al aumentar a 50 ms. el efecto sobre la media no se observa especial diferencia en los casos sin outliers entre las dos últimas transformaciones. Sin embargo, al comparar estas dos mismas transformaciones en el el respectivo caso 10 % de outliers, la norma residual parece estar más concentrada en torno a los valores centrales. Esto parece corresponderse con la gráfica de comparación con la distribución normal, en que se aprecia entre los mismos casos mejor ajuste a la normal en los valores bajos. Ligera disminución de los residuos de valor más alto.

Con efectos de 100 ms. sobre la media parece mejorar el ajuste de los mismos

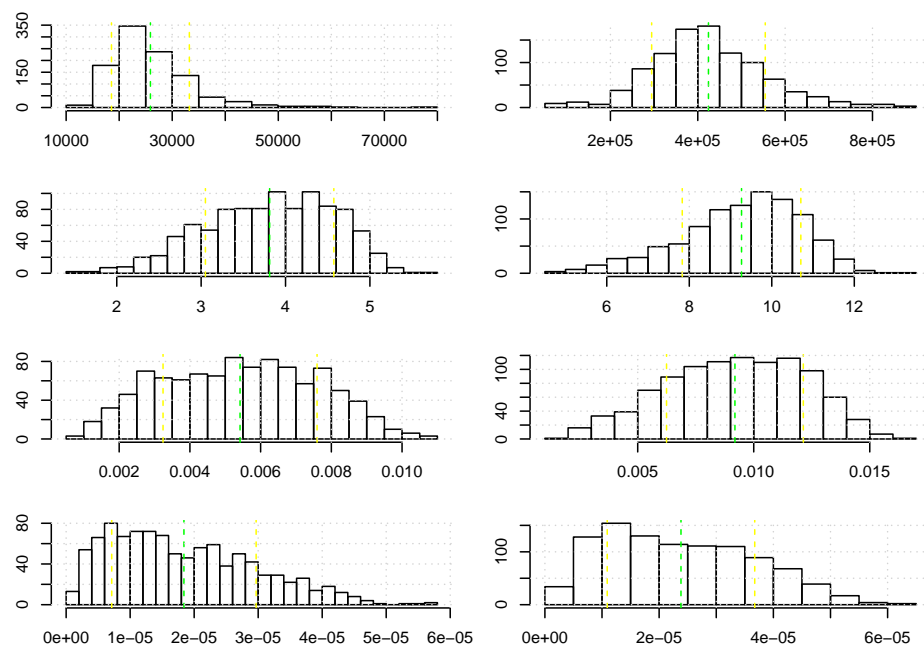


Figura 6.10: Sin efectos ni variabilidad intersujetos. Histograma de suma de residuos

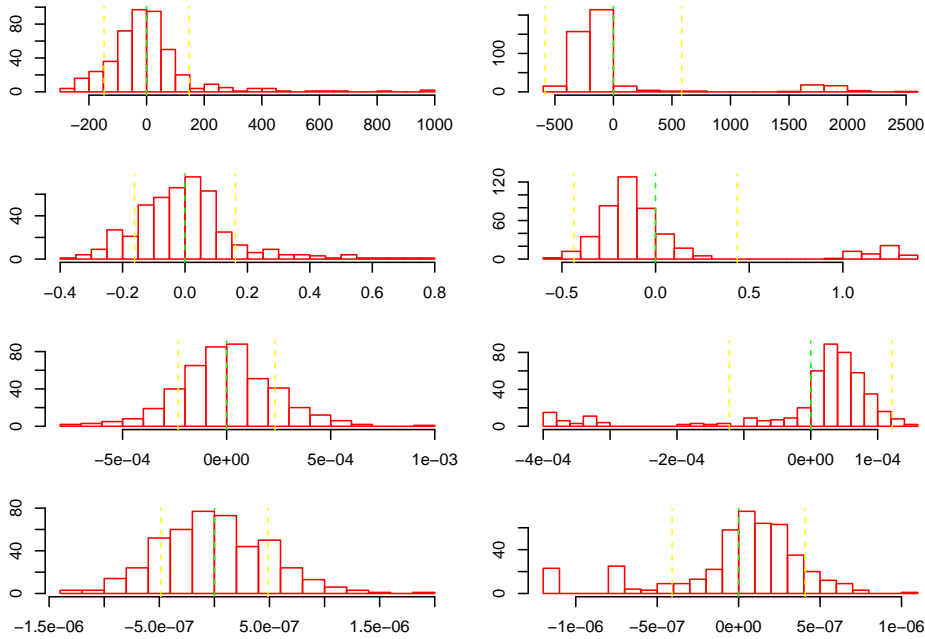


Figura 6.11: Sin efectos ni variabilidad intersujetos. Histograma.

a la distribución normal correspondiente, particularmente en los residuos de menor valor para el caso con un 10 % de outliers.

Con 1000 repeticiones del experimento la distancia a la correspondiente distribución normal parece agruparse más en los valores centrales. Se observa en este caso reducción de la media.

Con efectos de 100 ms. no se observa especial mejora en la distribución de los residuos. El *Test de Pearson* no muestra normalidad en la distribución de los residuos en ninguno de los casos estudiados.

### 6.3.3 Efecto sobre la cola y sin variabilidad entre sujetos

Es destacable como con un efecto de 50 ms. se obtiene mayor centralización de los residuos y mejor ajuste a la distribución normal de los mismos.

Con un efecto fijo, de hasta 100 ms., la transformación  $\frac{1}{TR^2}$  mejora el ajuste de los residuos a la recta Cuantil-Cuantil. Con una transformación fijada,

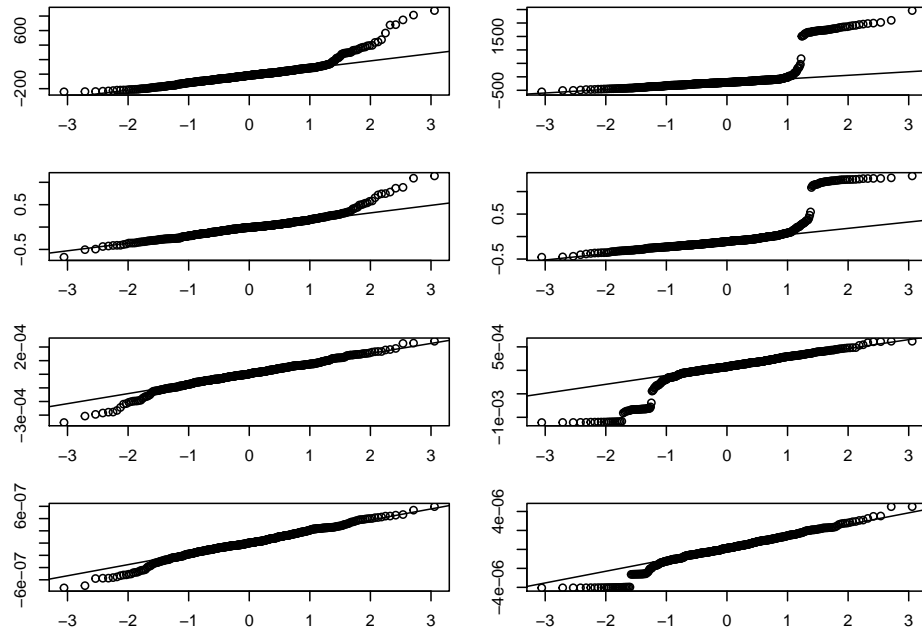


Figura 6.12: Efecto sobre la media 20 ms. QQ-Norm.

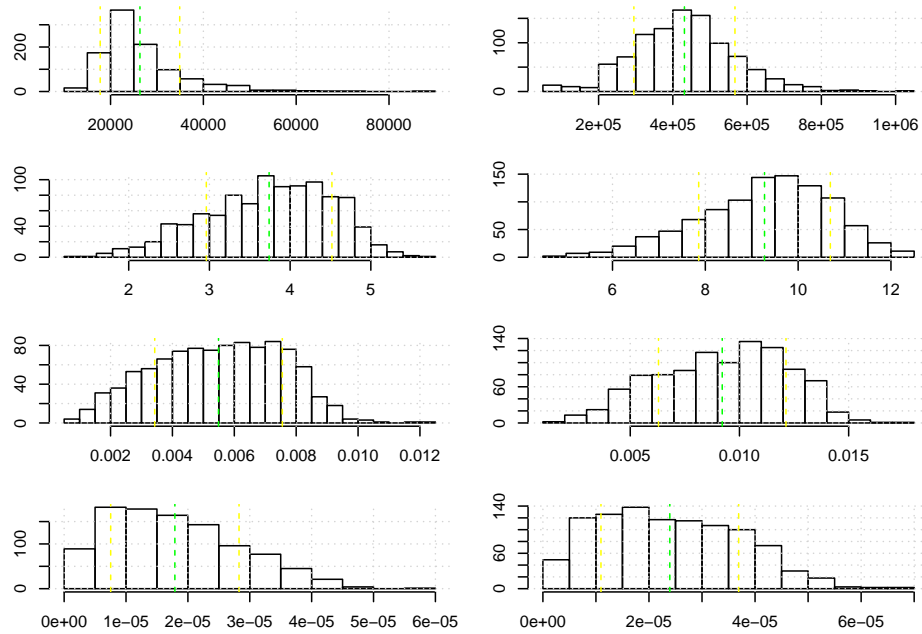


Figura 6.13: Efecto sobre la media 20 ms. Histograma de suma de residuos



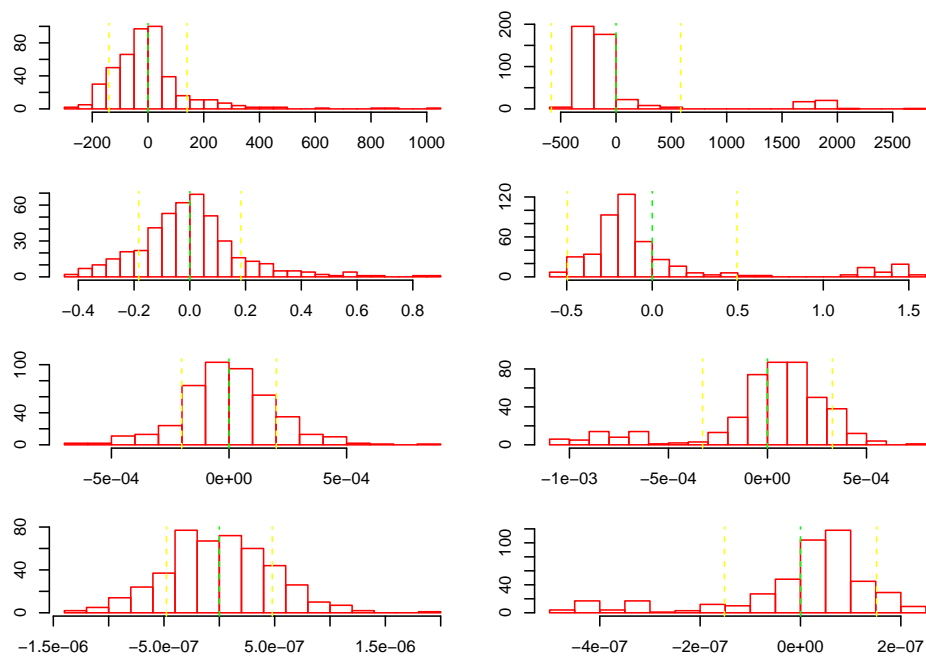


Figura 6.14: Efecto sobre la media 20 ms. Histograma.

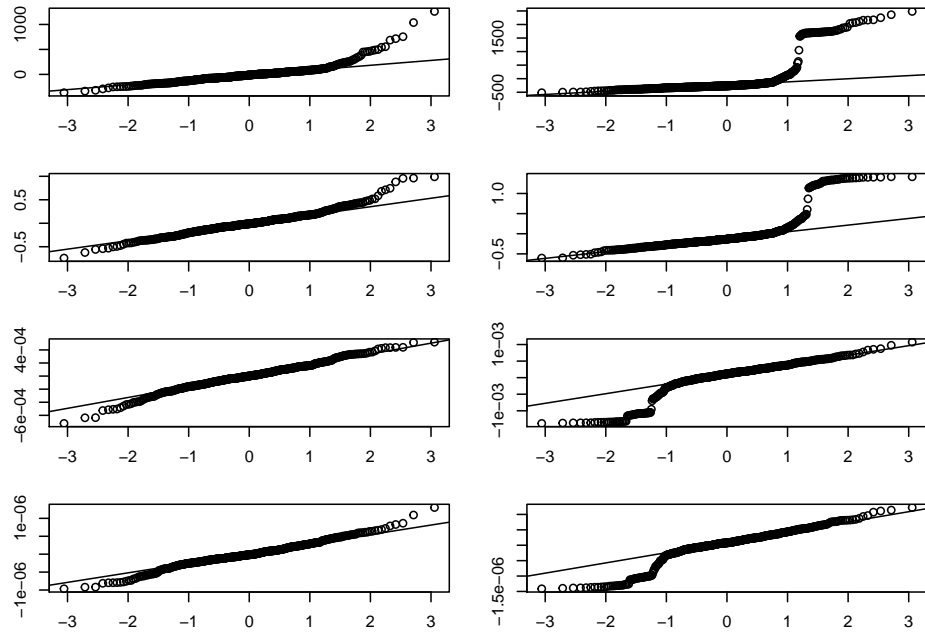


Figura 6.15: Efecto sobre la media 50 ms. QQ-Norm.

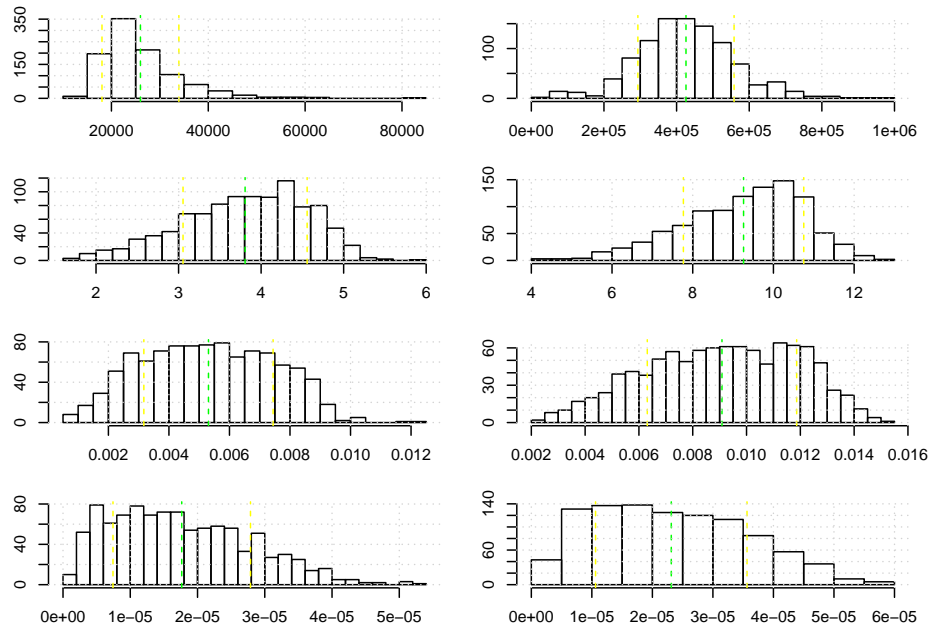


Figura 6.16: Efecto sobre la media 50 ms. QQ-Norm.

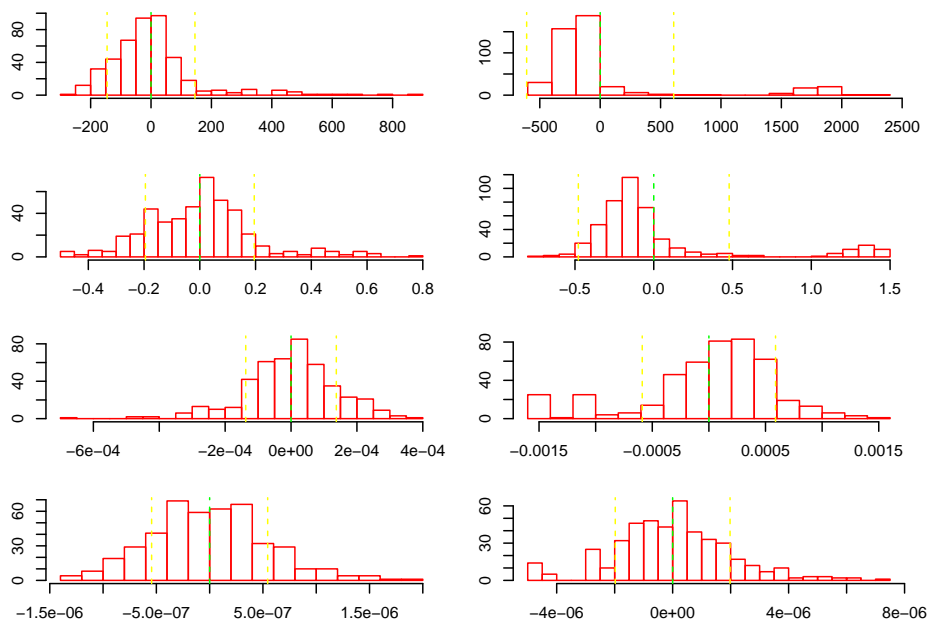


Figura 6.17: Efecto sobre la media 50 ms. Histograma.

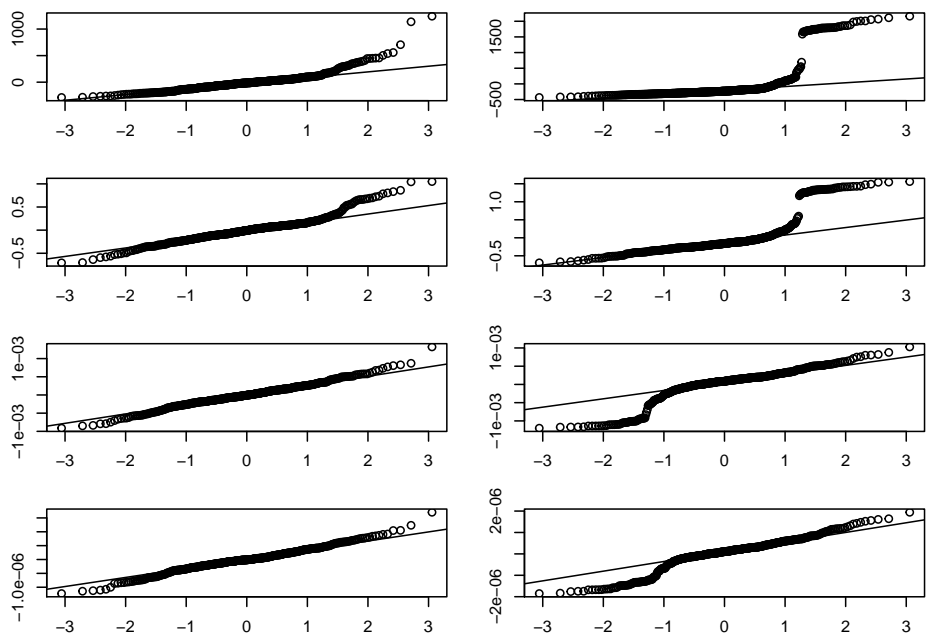


Figura 6.18: Efecto sobre la media 100 ms. QQ-Norm.

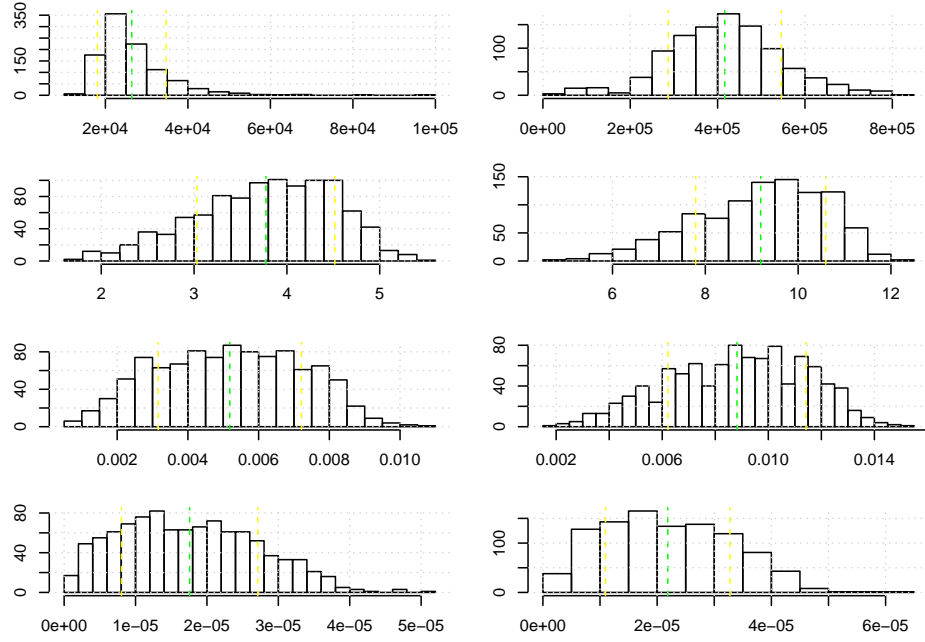


Figura 6.19: Efecto sobre la media 100 ms. Histograma de suma de residuos

aumentar el efecto sobre la cola reduce la diferencia entre los resícuos y la correspondiente recta Cuantil-Cuantil.

Frente a las demás situaciones analizadas en este apartado, aplicar un efecto grande sobre la cola normaliza la distribución de los residuos, siempre y cuando no se apliquen outliers. Frente al caso sin outliers, puede verse que al aplicar un 10 % de outliers (gráficos columna derecha) que ninguna de las transformaciones ofrece normalidad ni simetría pero que los casos  $\frac{1}{TR}$  y  $\frac{1}{TR^2}$  parecen invertir la asimetría de la distribución.

## 6.4 Conclusiones.

El análisis de potencia se basará en número de sujetos e ítems creciente comparando las transformaciones  $\frac{1}{t}$  y  $\frac{1}{t^2}$ . Recordemos que la distribución base es  $ExGauss(\mu, \sigma, \tau)$ .

- Si no se aplica ningún tipo de efecto sobre los parámetros  $\mu$  y  $\tau$  puede verse

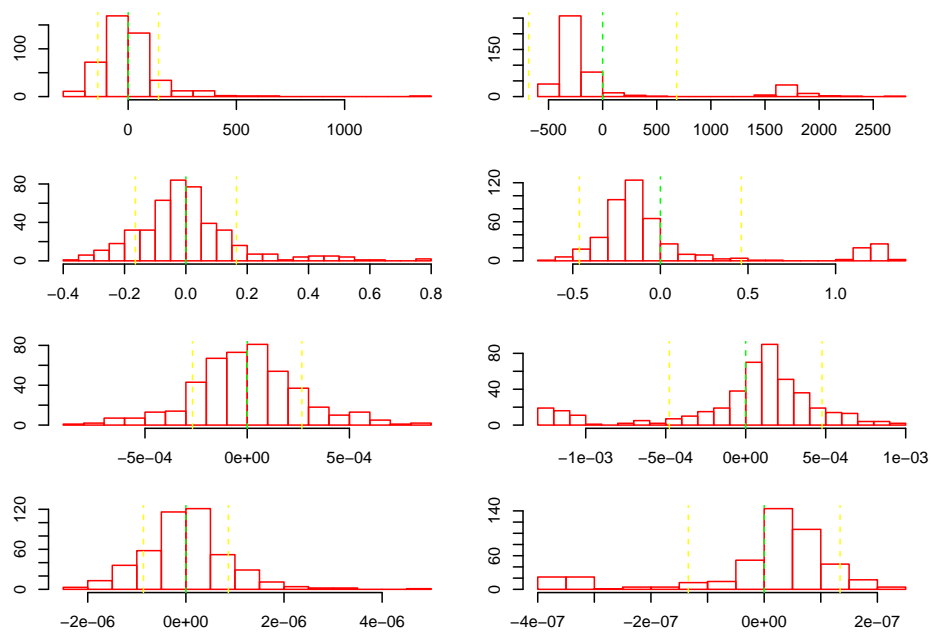


Figura 6.20: Efecto sobre la media 100 ms. Histograma.

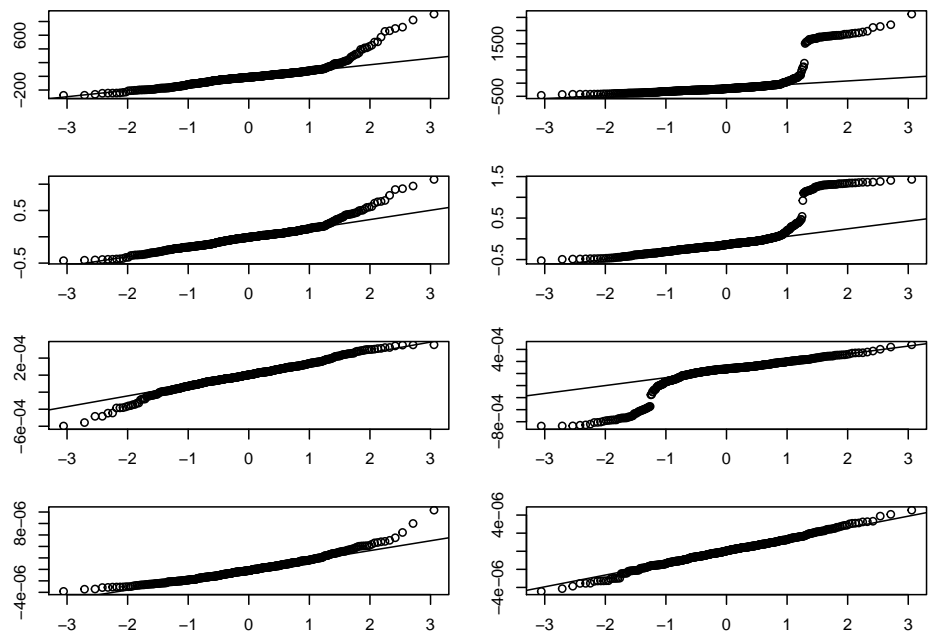


Figura 6.21: Efecto sobre la cola 20 ms. QQ-Norm.

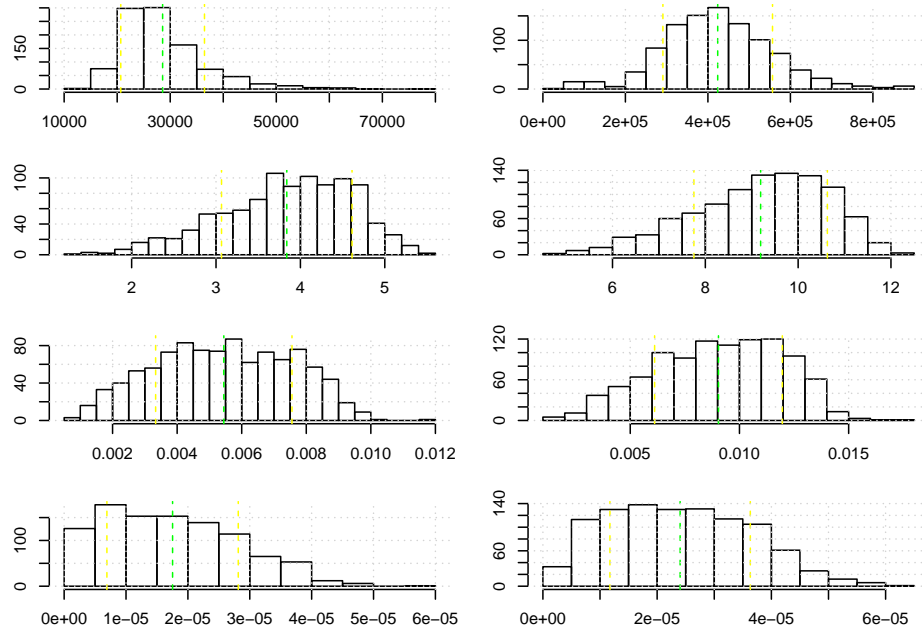


Figura 6.22: Efecto sobre la cola 20 ms. Histograma de suma de residuos.

que el error tipo I se mantiene entorno al 5 %. Disminuye ligeramente al aplicar la transformación  $\frac{1}{t^2}$ .

- Al aplicar efectos sobre la media, la transformación  $\frac{1}{t^2}$  reduce la diferencia entre los casos sin outliers y con un 10 % de outliers comparada con otras transformaciones previas. Esta reducción entre ambos casos deja de apreciarse con efectos sobre la media superiores a 50 ms.
- Los efectos sobre la cola, parámetro  $\tau$ , no ofrecen diferencias significativas entre los casos sin outlier y con un 10 % de outliers.
- Si se aplican diferentes efectos sobre los parámetros  $\mu$  y  $\tau$  puede observarse que al aumentar el efecto sobre alguno de dichos parámetros la diferencia entre las situaciones sin y con outliers disminuye considerablemente.

Para el análisis de residuos el desarrollo se basa en dos condiciones. La primera condición está basada en una distribución  $ExGauss(\mu = 400, \sigma = 40, \tau = 200)$  aplicada sobre 32 sujetos y 7 items por sujeto. La segunda condición, con el mismo número de sujetos e items por sujeto, se basa en la misma distribución sobre la que se han aplicado diferentes efectos sobre  $\mu$  o  $\tau$ .

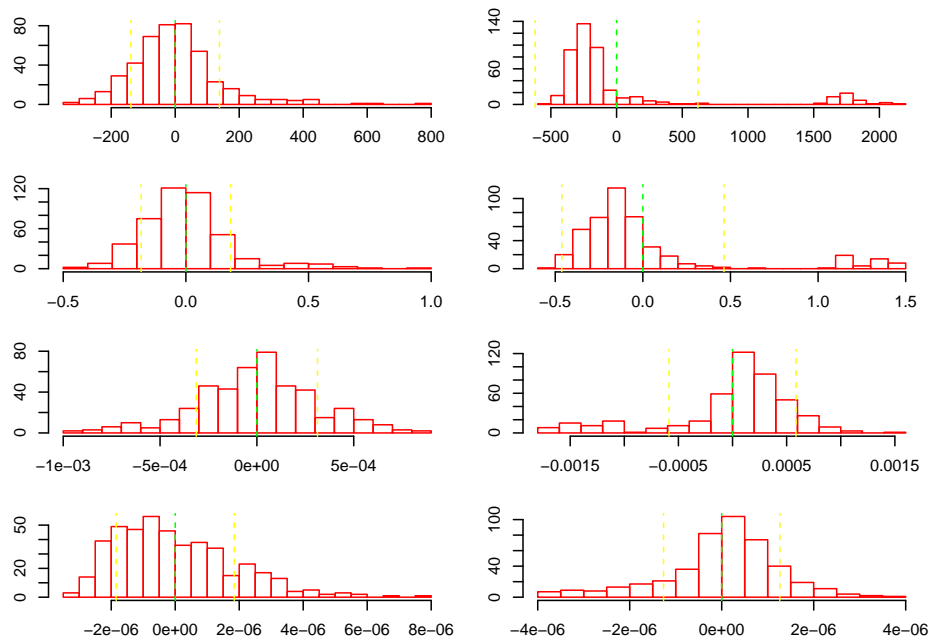


Figura 6.23: Efecto sobre la cola 20 ms. Histograma.

- Un incremento del efecto sobre  $\mu$  de hasta *100 ms.* mejora el ajuste de los residuos a la recta Cuantil-Cuantil.
- Aplicando efecto sobre la cola se observa mejor ajuste a la distribución normal. Esto se ha analizado con efectos sobre la cola de *20, 50 y 100 ms.*

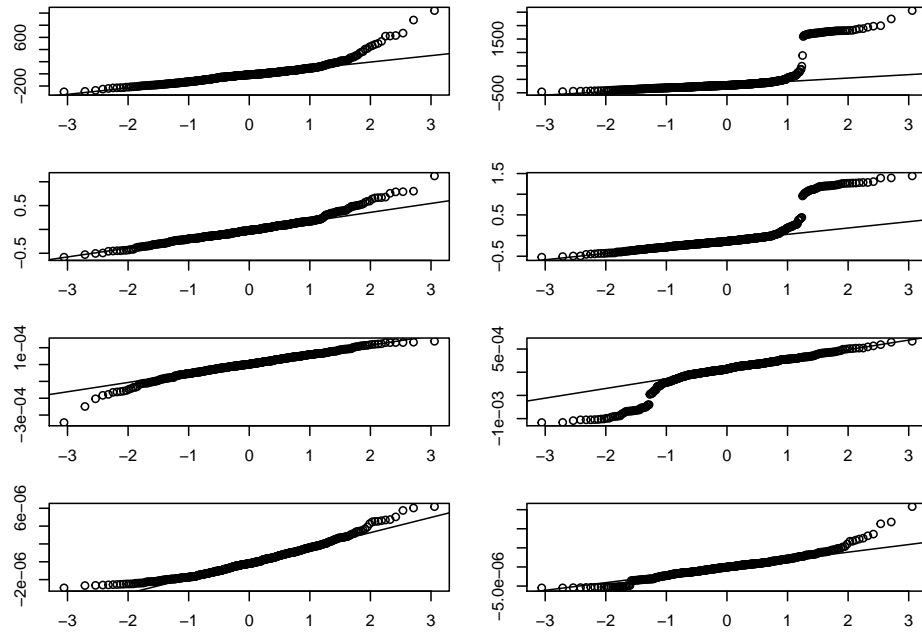


Figura 6.24: Efecto sobre la cola 50 ms. QQ-Norm.

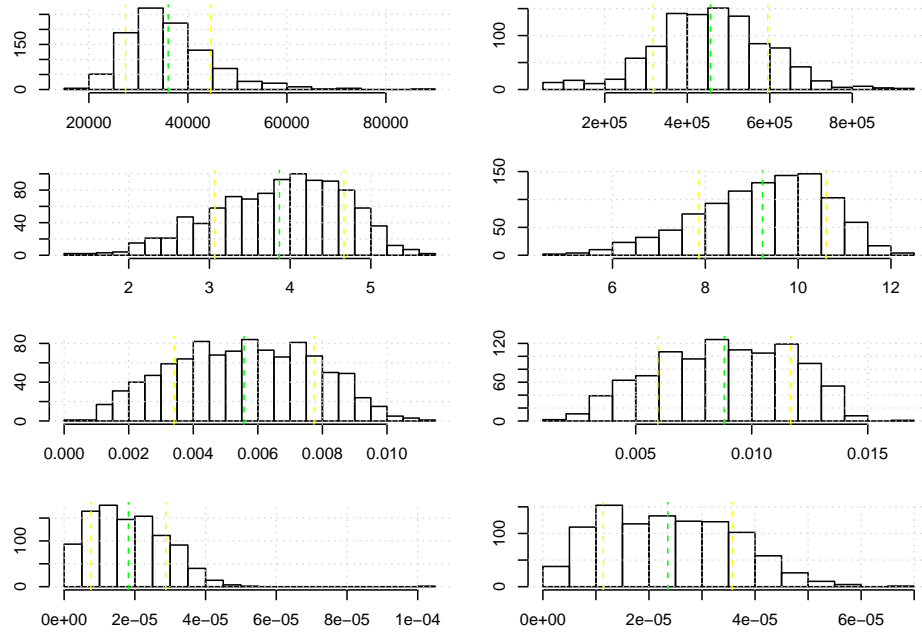


Figura 6.25: Efecto sobre la cola 50 ms. Histograma de suma de residuos.



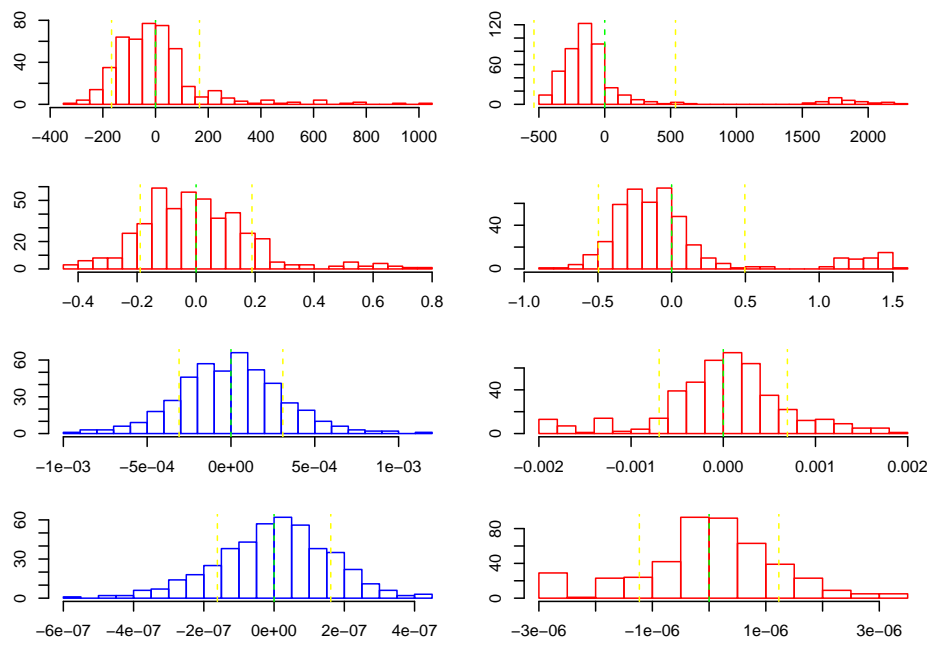


Figura 6.26: Efecto sobre la cola 50 ms. Histograma.

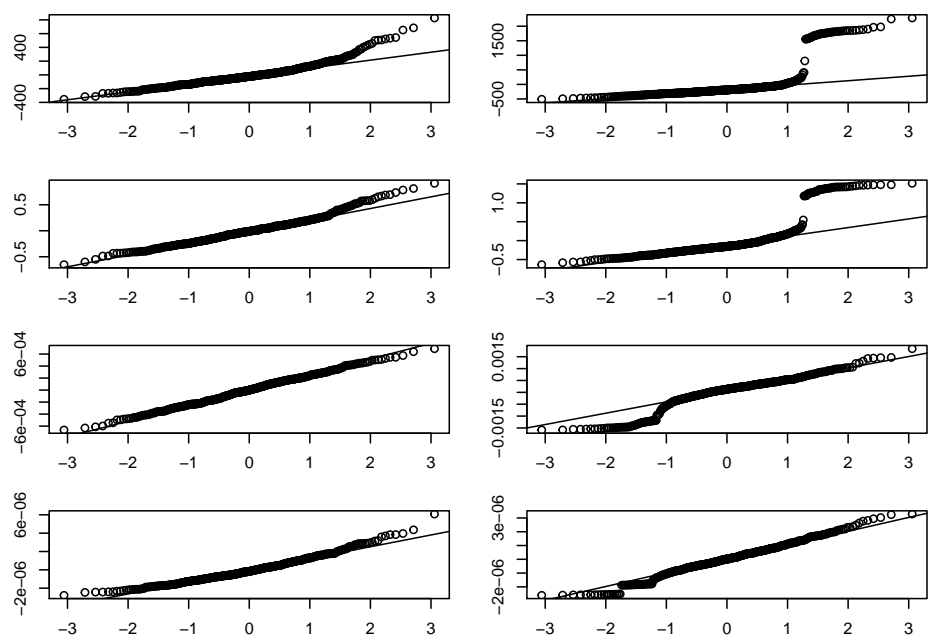


Figura 6.27: Efecto sobre la cola 100 ms. QQ-Norm.

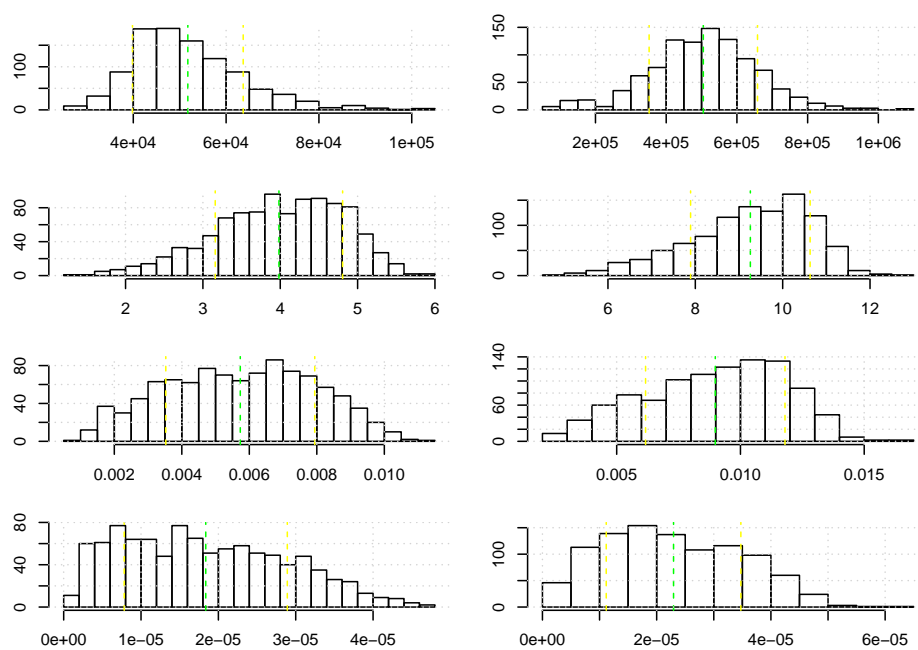


Figura 6.28: Efecto sobre la cola 100 ms. Histograma de suma de residuos.

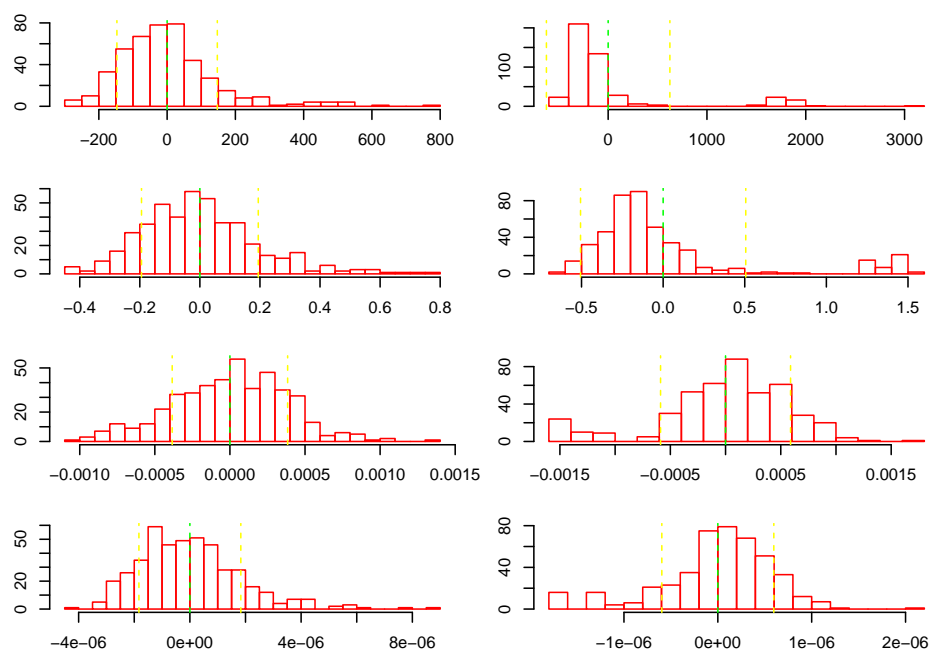


Figura 6.29: Efecto sobre la cola 100 ms. Histograma.

## Capítulo 7

# Tratamiento de Outliers en Tiempos de Respuesta de reconocimiento de palabras mediante transformaciones de datos y análisis de resultados mediante Modelos Lineales Generalizados Mixtos

### 7.1 Introducción y objetivos

En el trabajo de Ratcliff (1993) se completó un análisis de diferentes técnicas de tratamiento de Outliers para tiempos de respuesta, en adelante  $TR$ , en reco-

nocimiento de palabras con dos condiciones,  $n_S$  sujetos y  $n_I$  items por sujeto. Se evaluaron, entre otras, las transformaciones *No cut* (sin transformación),  $\log TR$  y  $\frac{1}{TR}$ . y los resultados fueron analizados mediante la técnica clásica *t-test*. Se concluyó que  $\frac{1}{TR}$  era el método que ofrecía mejores resultados. En un desarrollo previo al presente, se replicaron los resultados de potencia de Ratcliff (1993) y se incorporó la transformación  $\frac{1}{TR^2}$  al análisis de potencia *t-test*.

El objetivo del desarrollo que sigue es emplear las mismas condiciones iniciales que en Ratcliff (1993) y realizar también un análisis de bondad de los diferentes métodos de tratamiento de outliers desde el punto de vista de un *Modelo Generalizado Lineal Mixto*, a modo de lo planteado en Baayen, Davidson & Bates (2008) y Barra, Levyb, Scheepers & Tilyc (2013). Es decir, en este apartado se tratarán los principales métodos de tratamiento de outliers así como la nueva transformación  $\frac{1}{TR^2}$ , desde el punto de vista del análisis de un *Modelo Generalizado Lineal Mixto*.

Pueden repasarse conceptualmente los *Modelos Lineales* y *Modelos Lineales Generalizados Mixtos* en los trabajos Bodo (2013b) y Bodo (2013a).

Empleando, al igual que en Ratcliff (1993), una distribución *ExGauss* ( $\mu = 400, \sigma = 40, \tau = 200$ ) y tomando una muestra de 1000 items / lecturas representaremos la muestra aleatoria con las cuatro posibles transformaciones *Nocut* (sin transformación),  $\log TR$ ,  $\frac{1}{TR}$  y  $\frac{1}{TR^2}$ .

La generación de datos y preprocesado con las diferentes transformaciones puede verse en la sección previa específicamente dedicada a esta cuestión.

Nótese que en el ámbito de la Psicología Cognitiva es frecuente partir de datos procedentes de lecturas de *ERP*. Estas lecturas miden tiempos de reacción frente a diferentes estímulos en base a la respuesta neurofisiológica. Véase a modo de ejemplo, Vergara-Martínez, Perea, Gómez and Swaab (2013).

## 7.2 Modelo Lineal frente a Modelo Lineal Generalizado Mixto

Los conceptos básicos de ambos modelos pueden repasarse en los trabajos de Bodo (2013b) y Bodo (2013a), y se resumen esencialmente en las secciones que siguen en este mismo apartado. Aunque el objetivo específico de esta sección del

trabajo es el estudio mediante el uso de un *Modelo Lineal Generalizado Mixto* se repasarán brevemente las bases del *Modelo Lineal* para entender posteriormente la diferencia de criterio entre ambas opciones.

### 7.2.1 Modelo Lineal

Si estudiamos dos características o variables aleatorias ( $x$  e  $y$ ) en el mismo grupo de sujetos obtenemos una relación

$$y \sim x$$

Podemos decir que “ $y$  depende de  $x$ ”, o en términos matemáticos clásicos  $x$  es la variable independiente e  $y$  es variable dependiente. Una relación de este tipo sólo es válida en una situación ideal en que la variable  $y$  se considera exclusivamente dependiente de  $x$ . Podríamos incorporar al modelo o ecuación anterior un componente aleatorio o probabilístico, que expresaremos  $\epsilon$  que explique la parte no fijada del segundo miembro de la ecuación, de forma que

$$y \sim x + \epsilon$$

Podríamos añadir al modelo lineal anterior otras variables independientes  $x_1, x_2, \dots$  que determinaran la variable  $y$ , además del componente aleatorio  $\epsilon$ , de forma que

$$y \sim x_1 + x_2 + \dots + \epsilon$$

Todos los casos anteriores son modelos lineales, es decir,  $y$  queda determinado mediante una combinación lineal de  $x_1, x_2, \dots$ .

Veamos ahora las condiciones que determinan la linealidad de un modelo.

1. **Linealidad.** Es suficiente representar el gráfico de predicciones frente al de los residuales para observar si los residuos siguen un patrón lineal o si los mismos residuos siguen un patrón no linealidad.
2. **Ausencia de colinealidad.** Cuando dos efectos fijos o predictores están relacionados entre sí diremos que existe colinealidad entre ambos. Si se produce esta situación, la interpretación de los resultados se complica dentro del marco base de un modelo predictivo lineal. Debe plantearse el experimento inicial evitando poder incurrir en situaciones de colinealidad.

3. **Homocedasticidad o ausencia de heterocedasticidad.** Un modelo presenta homocedasticidad cuando la varianza del error para los diferentes efectos fijos es constante a lo largo de las observaciones. Tanto la colinealidad (ausencia de) como la homocedasticidad pueden tratarse controlándose aplicando diferentes transformaciones sobre los datos, tema de estudio en concreto del presente trabajo.
4. **Normalidad de los residuos.** La normalidad de los residuos es una cuestión cuya importancia para determinar la linealidad del modelo es controvertida entre los diferentes modelos. Particularmente, el texto de Gellman & Hill (2007) no recomienda dicho estudio de normalidad de residuos. Para su estudio dispondríamos de dos opciones: *a)* histograma de residuos (`hist`), y *b)* Q-Q plot que representa la regresión lineal de los propios residuos obtenidos del modelo lineal de ajuste previo (`qqnorm`).
5. **Ausencia de puntos de influencia.** Puede utilizarse en R la función `DFBeta` para evaluar el peso específico de cada efecto dentro del modelo lineal, es decir, la variación del correspondiente coeficiente si se descartara el uso de esta variable en el modelo final.
6. **Independencia.** Se ha demostrado en diferentes trabajos y ámbitos de estudio que no es infrecuente incumplir la independencia estadística de las lecturas. Sirvan como ejemplos los trabajos de Feeberg & Lucas (2009) en Psicología, Lazic (2010) en Neurociencia o Winter (2011) en Neurolingüística.

### 7.2.2 Modelo Lineal Generalizado Mixto

De forma similar a un modelo lineal, este tipo de planteamiento se basa en una expresión del tipo

$$y \sim x + \epsilon$$

, siendo  $x$  el efecto fijo o variable independiente y  $\epsilon$  el término de error aleatorio o componente escolástico que justifica el desvío de los datos en relación a la predicción del modelo.

- Sobre la base del planteamiento anterior se añade el hecho de que la variable independiente o efecto fijo  $x$  pudiera tener un comportamiento categórico. Es decir, se tendrían lecturas del mismo tipo de datos pero en diferentes circunstancias o condiciones. Por ejemplo, en casos aplicados



a la lectura, registros formal / informal en Winter & Grawunder(2012), minúscula / mayúscula en Perea, Jiménez, Talero & López-Cañada (2014) o Perea, Rosa & Marcet (2017).

- Además, cada sujeto presentará diferentes lecturas bajo cada condición, como viene siendo habitual en los estudios de neurolingüística. Las lecturas obtenidas para cada sujeto y condición vendrán determinadas por una serie de items.

Los dos elementos señalados hacen que los diferentes conjuntos de datos, diferenciados por condición y sujeto, no sean independientes y se incumplan las premisas del modelo lineal. Es decir, esencialmente cada sujeto sometido al experimento en una de las condiciones posibles proporcionará al conjunto de datos final una serie de características que lo harán único en sí mismo.

Utilizando la notación tradicional del software `R-Project` (2006), el nuevo planteamiento del modelo será del tipo

$$y \sim x + (1|subj) + (1|cond) + \epsilon$$

, de forma que cada sujeto independiente presenta diferentes lecturas por sujeto.

Las premisas de los modelos lineales recordadas en el apartado previo siguen siendo válidas en el caso de modelos mixtos. Mención aparte debería hacerse en lo relativo a la independencia, ya que como se ha aclarado previamente, un sujeto presenta diferentes lecturas / items por lo que los resultados de cada lectura en un mismo sujeto no son independientes.

### 7.3 Planteamiento del problema y Modelo Lineal Generalizado Mixto

Consideremos 2 condiciones,  $n_S$  sujetos y  $n_I$  items. Bajo la primera condición pueden por tanto considerarse  $n_S$  muestras de  $n_I$  items independientes e idénticamente distribuidas en base a una distribución  $ExGauss(\mu, \sigma, \tau)$ . La generación aleatoria de datos en los experimentos que siguen, en base a una distribución  $ExGauss$  se planteó inicialmente en el trabajo de Ratcliff (1979).

Si los  $n_S$  sujetos se someten a los mismos  $n_I$  items debemos tener en cuenta la

variabilidad entre items dentro de un mismo sujeto y también la variabilidad entre sujetos para el mismo item. Particularmente, partiendo de  $ExGauss(\mu, \sigma, \tau)$ , para cada sujeto  $\mu \sim ExGauss(\mu, \sigma, \tau)$ ,  $\sigma \sim N(\sigma, \frac{\sigma}{2})$  y  $\tau \sim N(\tau, \frac{\tau}{2})$ .

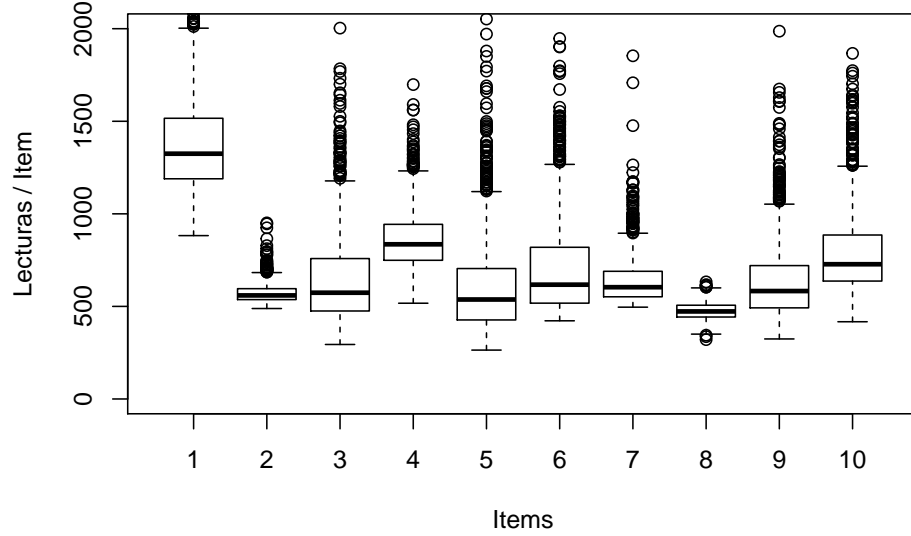


Figura 7.1: Distribución ExGauss de 10 items

Las  $n_S$  muestras del segundo sujeto emplean efectos sobre los tres parámetros de la distribución *ExGauss* original dados por  $\mu_{ef}$ ,  $\sigma_{ef}$  y  $\tau_{ef}$ .

De este modo, podemos considerar cada *TR* analizado como perteneciente a un posible sujeto, item y condición. El análisis de datos ya no es por tanto convencional, como en Ratcliff (1993) sino que emplearemos un *Modelo Lineal de Efectos Mixtos*, similarmente a Baayen et al. (2008).

Mientras que en el modelo de Baayen et al. (2008) se empleaba el modelo

$$RT \sim SOA + (1|Item) + (1|Subj)$$

, en el presente trabajo nos centraremos en el modelo

$$RT \sim SOA + (1 + SOA|Item) + (1 + SOA|Subj)$$

, ambos expresados en el formato del software R-Project (2006).

El análisis computacional se hará empleando el software estadístico

**R-Project** (T. R. C. Team, 2006) , embebido dentro de su *FrontEnd R-studio* (2015), que se ha empleado para la redacción del presente documento, conjuntamente con el sistema de procesamiento de textos **TeX** (1984). La librería particular empleada es **lme4** (2015), de uso específico para Modelos Generalizados Lineales Mixtos. Las múltiples simulaciones estadísticas que han permitido generar las muestras respectivas y analizar los correspondientes resultados se han ejecutado en el entorno del *Supercomputador Lluís Vives* de la *Universitat de València*, cuya información detallada puede consultarse en la web del *Centro de Cálculo* de la *Universitat de València* («WIKI del Servicio de Informática SiUV», 2008). Ha sido necesario el uso de trabajos en paralelo y uso de multiprocesador en los trabajos computacionales dado el gran volumen de cálculo y el tiempo de uso limitado por defecto a 96 horas por trabajo y usuario.

En el apartado siguiente, correspondiente al análisis de potencia mediante Modelos Generalizados Lineales Mixtos puede verse el código **R** para la generación de muestras y análisis de potencia.

Los datos se organizan en una estructura *Data Frame* de 4 columnas del tipo

**#|Subj|Item|SOA|TR|**

tal y como se muestra en Baayen et al. (2008). La primera columna **Subj** indica el sujeto (valores de 1 a  $n_S$ ), **Item** señala el ítem (valores de 1 a  $n_I$ ), la columna **SOA** indica la condición (valores 1 y 2) y finalmente la columna **TR** con los Tiempos de Respuesta o las correspondientes transformaciones de los mismos. Por tanto, estamos hablando de una tabla de datos con  $n_S \times n_I \times 2$  entradas.

En las simulaciones que siguen se emplearán 32 sujetos (**nsujetos=32**), 7 ítems (**nitems=7**) en cada caso. En concreto, estamos por tanto hablando de una tabla de  $32 \times 7 \times 2 = 448$  entradas. Esto se hace tomando como referencia el trabajo de Baayen et al. (1993).

Las cuatro transformaciones posibles de un *TR* son:  $t$  (sin transformación / No cut),  $\log(t)$  (logaritmo neperiano),  $\frac{1}{t}$  (inversa) y  $\frac{1}{t^2}$  (inversa de la potencia 2). Estas funciones transformación son positivas y crecientes si  $t > 0$  y además

$$\frac{1}{t^2} < \frac{1}{t} < \log(t) < t, \forall t \in ]1, +\infty[$$

Matemáticamente, consideraremos los *TR* generados en base a una distribución ExGauss y expresados en *ms*. Sean  $\mu$ ,  $\sigma$  y  $\tau$  los parámetros que definen la media, desviación típica y cola de la distribución ExGauss  $T$  generada como suma de las distribuciones  $Ex(\tau)$  y  $N(\mu, \sigma)$ . Conocidas las distribuciones de ambas, la función de densidad de la *ExGauss* resultante es de la forma

$$f(t|\mu, \sigma, \tau) = \tau e^{-\tau t} + \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{1}{2} \left(\frac{t-\mu}{\sigma}\right)^2}, t > 0$$

. Por el mismo motivo, podemos obtener la esperanza y varianza de la distribución ExGauss de los *TR*. Entendamos por  $T$  la variable aleatoria de los *TR*.

$$E(T) = \frac{1}{\tau} + \mu, Var(T) = \frac{1}{\tau^2} + \sigma^2, t > 0$$

Particularmente, la distribución *ExGauss* básica de los experimentos que siguen tiene por parámetros  $\mu = 400$ ,  $\sigma = 40$  y  $\tau = 200$ .

## 7.4 Análisis de potencia

Se presenta en primer lugar el código empleado para la generación de muestras y análisis correspondiente de potencia.

```
outliers_glme_pow<-function(mu=400,sigma=40,tau=200,mu.ef=30,sigma.ef=0,
tau.ef=0,p.outlier=0.1,nsujetos=32,nitems=7,nrep=1000,metodo="nocut")
{
  contador<-0;

  for(p in 1:nrep)
  {
    # Generación de una muestra aleatoria

    A<-matrix(nrow=1,ncol=4);

    medias_base<-rnorm(n=nitems,mean=mu,sd=sigma)+rexp(n=1,rate=1/tau);

    for(i in 1:nsujetos)
    {
      A1<-matrix(nrow=nitems*2,ncol=4);
```

```

muestra1<-rep(x=0,times=nitems);
muestra2<-rep(x=0,times=nitems);

desviaciones_base<-rnorm(n=nitems,mean=sigma,sd=0.5*sigma);
colas_base<-rnorm(n=nitems,mean=tau,sd=0.5*tau);

for(j in 1:nitems)
{
  # Muestra aleatoria Condición 1
  muestra1[j]<-rnorm(n=1,mean=medias_base[j],sd=desviaciones_base[j])+
  rexp(n=1,rate=1/colas_base[j]);

  # Muestra aleatoria Condición 2
  muestra2[j]<-rnorm(n=1,mean=medias_base[j]+mu.ef,sd=desviaciones_base[j]+
  sigma.ef)+rexp(n=1,rate=1/(colas_base[j]+tau.ef));
}

# Generamos una Un(0,1) y a los elementos de la muestra cuyo valor
# uniforme aleatorio tenga valor menor a p.outlier le sumamos 2000
# para simular un % de outliers

aleatorio<-runif(n=nitems,min=0,max=1);
muestra1[which(aleatorio<p.outlier)]<-
muestra1[which(aleatorio<p.outlier)]+2000;

aleatorio<-runif(n=nitems,min=0,max=1);
muestra2[which(aleatorio<p.outlier)]<-
muestra2[which(aleatorio<p.outlier)]+2000;

# Transformaciones sobre los datos de las muestras generadas
# en cada condición

if(metodo=="log")
{
  muestra1<-log(x=muestra1);
  muestra2<-log(x=muestra2);
}

```

```

}
if(metodo=="inv")
{
  muestra1<-1/muestra1;
  muestra2<-1/muestra2;
}
if(metodo=="inv2")
{
  muestra1<-1/(muestra1^2);
  muestra2<-1/(muestra2^2);
}

# Se añade dato a la tabla DataFrame de valores

for (k in 1:nitems)
{
  A1[k,]<-c(i,k,1,muestra1[k]);
  A1[k+nitems,]<-c(i,k,2,muestra2[k]);
}

A<-rbind(A,A1);
}

# Eliminamos primera fila con títulos de columnas.

A<-A[-1,];

# Planteamiento del GLME y obtención de la potencia

datos<-data.frame(Subj=A[,1],Item=A[,2],SOA=A[,3],RT=A[,4]);

lme0<-lmer(RT~SOA+(1+SOA|Item)+(1+SOA|Subj),data=datos,REML=F);
lme1<-lmer(RT~1+(1+SOA|Item)+(1+SOA|Subj),data=datos,REML=F);

anova_results<-anova(lme0,lme1);

```

```

pvalor<-anova_results[8]$Pr[2];

if(pvalor<0.05)
{
  contador<-contador+1;
}
}

# Recuento de casos favorables y retorno de proporción

return((contador/nrep)*100);
}

```

El *t-test* no es un método válido para estudiar la significatividad de Modelos lineales generalizados mixtos. Como alternativa, en base los trabajos de Pinheiro & Bates (2000) y Bolker, Brooks, Clark, Geange, Poulsen, Stevens & White (2009) se comparan dos modelos del tipo

$$lme0 < -lmer(RT\ SOA + (1 + SOA|Item) + (1 + SOA|Subj), data = datos, REML = F)$$

y

$$lme1 < -lmer(RT\ 1 + (1 + SOA|Item) + (1 + SOA|Subj), data = datos, REML = F)$$

, aplicándose sobre el resultado de ambos un test **ANOVA**.

Basándonos en el algoritmo anterior de cálculo de potencia, en cada sección de este apartado se muestra un gráfico múltiple en que se indica de arriba a abajo el análisis para 10, 20, 30 y 40 sujetos. De izquierda a derecha 10, 20, 30 y 40 items. En cada gráfico se muestra el resultado para cada tipo de transformación sin outliers y con un 10 % de outliers.

### 7.4.1 Análisis error tipo I

Sin efecto sobre ninguno de los parámetros. Mismas condiciones que en Ratcliff (1993).

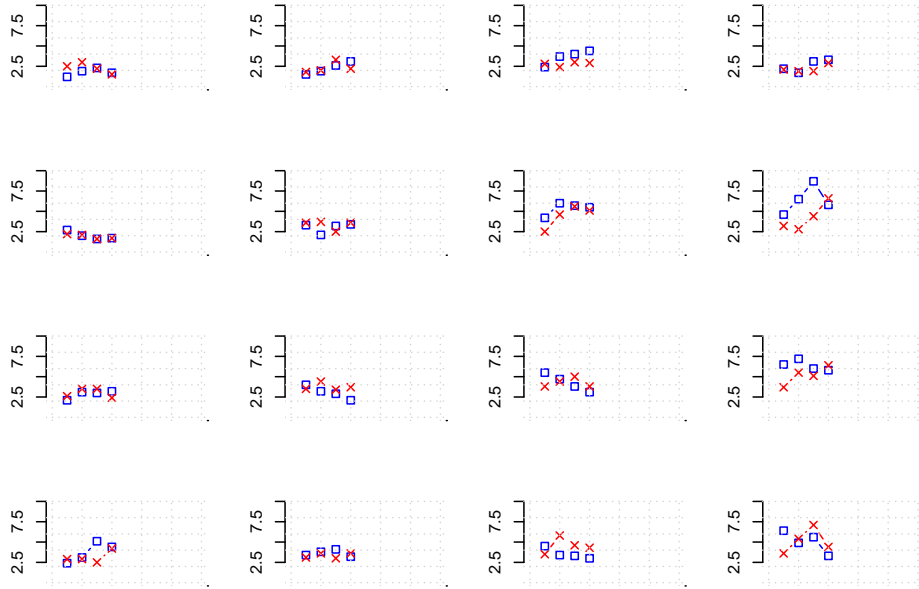


Figura 7.2: Análisis de falsos positivos

No aplicando efecto sobre ninguno de los parámetros de la distribución *Ex-Gauss* se observa que la proporción de falsos positivos se mantiene alrededor del 5 %.

En trabajos previos, procedentes de los ámbitos de la Ecología, Schielzeth & Forstmeier (2009), y la Psicolingüística, Barra et al. (2013), se muestra en base a simulaciones que en el caso de modelos mixtos el Error de Tipo I se mantiene en tasas altas siempre que no se tenga en cuenta el componente aleatorio.

### 7.4.2 Efectos sobre la media

Con pocos 10 sujetos y 10 items, no se observan diferencias significativas entre los casos sin y con outliers. La diferencia entre ambos casos aumenta al incrementar número de sujetos y/o items. Con un número elevado de sujetos



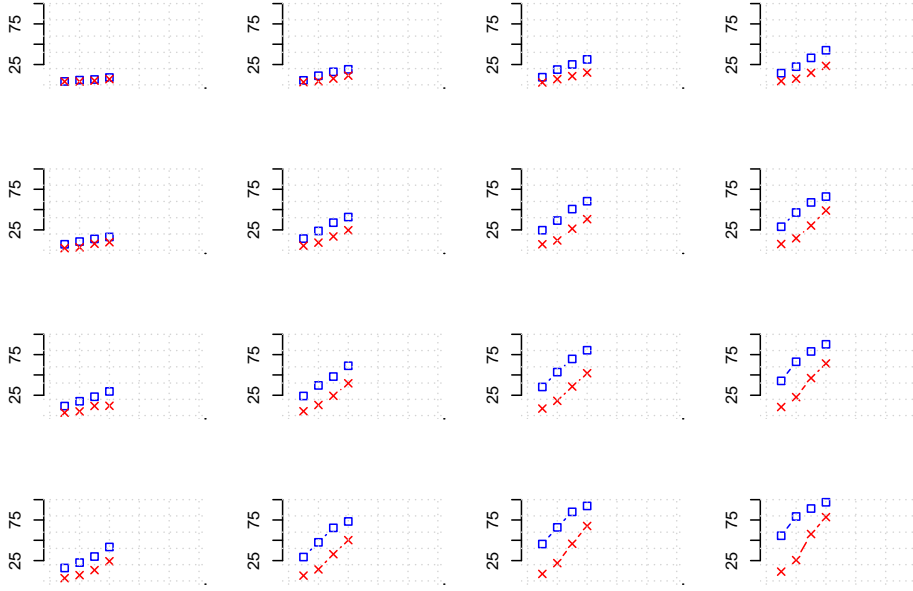


Figura 7.3: Efecto de 20 ms. sobre la media.

e items, observamos una disminución moderada de la diferencia entre los casos sin outliers y con outliers para la transformación  $\frac{1}{T^2}$  si la comparamos con  $\frac{1}{T}$

Con un efecto de 50 ms., al aumentar el número de sujetos e items, la diferencia entre los casos sin outliers y con un 10 % de outliers se ve casi anulada tanto en el caso  $\frac{1}{TR}$  como  $\frac{1}{TR^2}$ .

De forma similar al caso con un efecto sobre la media de 50 ms., al aumentar dicho efecto hasta los 100 ms., la diferencia entre los casos sin outliers y con un 10 % de outliers. La diferencia frente al caso anterior es que esta diferencia se ve casi anulada ya para número de sujetos e items pequeño y para todas las transformaciones panteadas.

### 7.4.3 Efectos sobre la cola

Un efecto pequeño sobre la cola de 20 ms. hace casi imperceptible la diferencia entre los casos sin outliers y con un 10 % de outliers. En todos los casos, particularmente con número de sujetos e items grande, se aprecia como la transformación  $\frac{1}{T^2}$  ofrece disminución de la diferencia entre ambos casos frente a

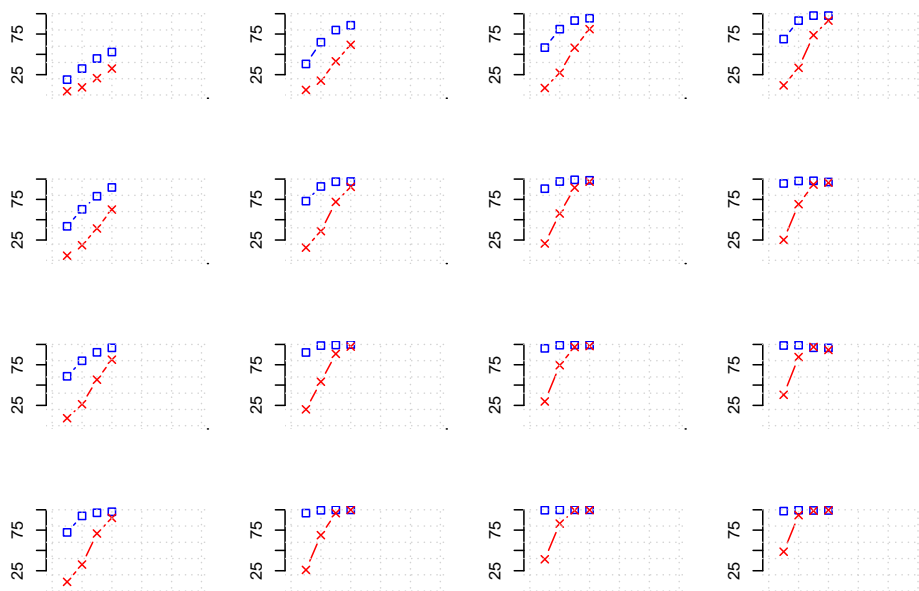


Figura 7.4: Efecto de 50 ms. sobre la media

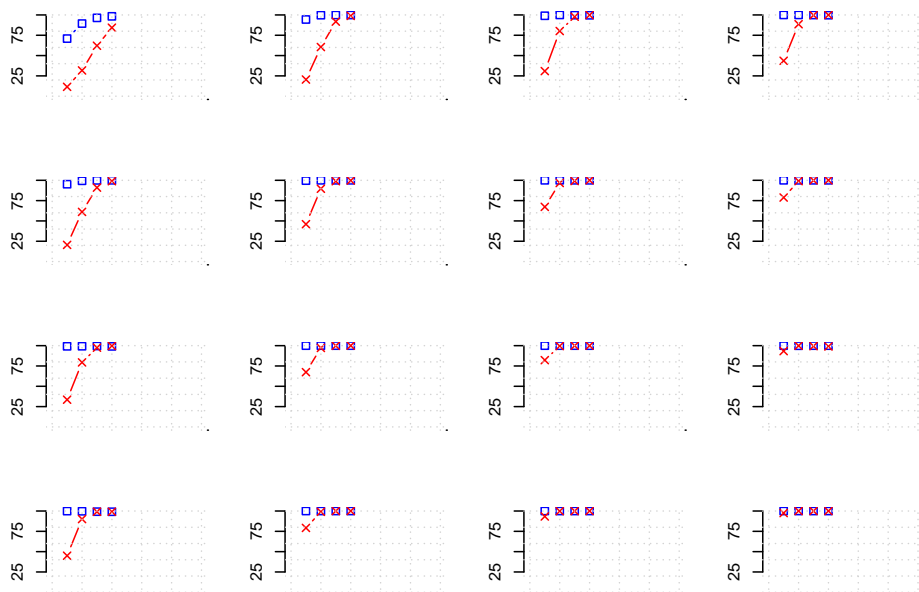


Figura 7.5: Efecto de 100 ms. sobre la media

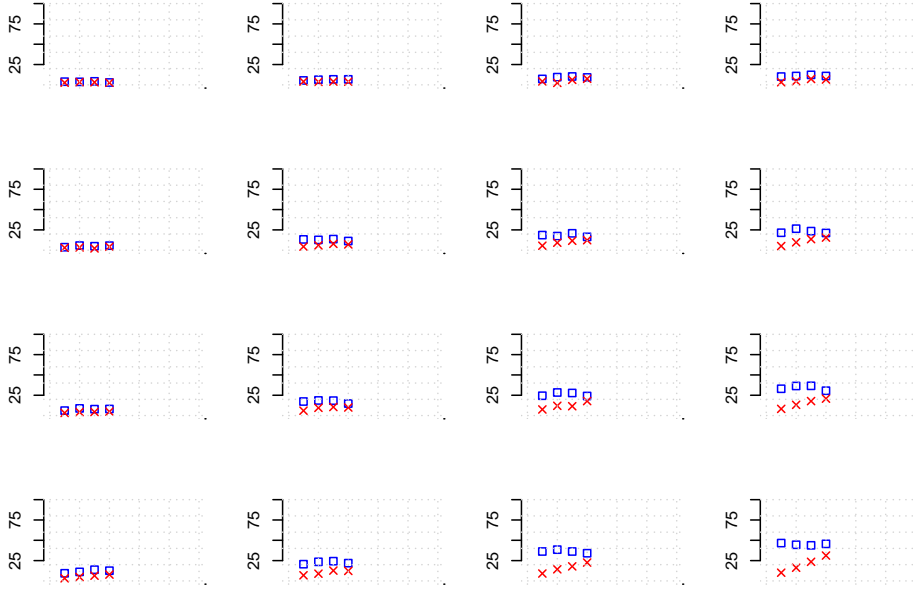


Figura 7.6: Efecto de 20 ms. sobre la cola

cualquier otra transformación.

Al aumentar hasta *50 ms.* el efecto sobre la cola, la diferencia entre los casos sin outliers y con un *10 %* de outliers, que es grande con los métodos anteriores se reduce notablemente al aplicar la transformación  $\frac{1}{TR^2}$ .

Al igual que al aplicar efectos grandes sobre la media, si se aplica un efecto grande sobre la cola, la diferencia entre los casos sin outliers y con un *10 %* de outliers se ve casi anulada para toda transformación, incluyendo para pocos sujetos e items.

## 7.5 Análisis de residuos

La pretensión de esta sección es el análisis de los residuos generados con cada tipo de transformación al aplicar el modelo generalizado lineal mixto.

Recordemos, según se indicó en el planteamiento del modelo, que partiremos de *32* sujetos representados por *7* items por sujeto y *2* condiciones, acorde a las condiciones de Ratcliff (1993). Es decir,  $32 \times 2$  simulaciones *ExGauss* de

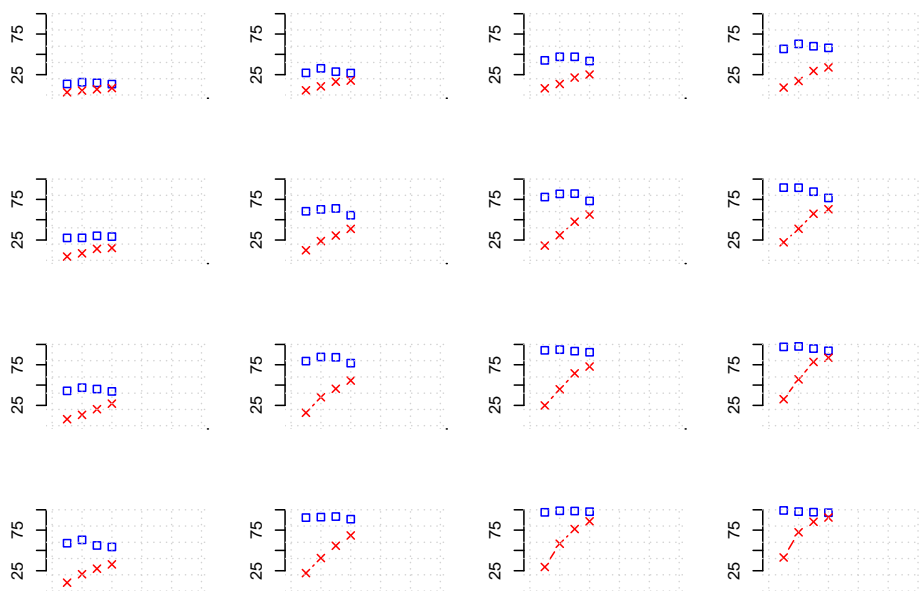


Figura 7.7: Efecto de 50 ms. sobre la cola

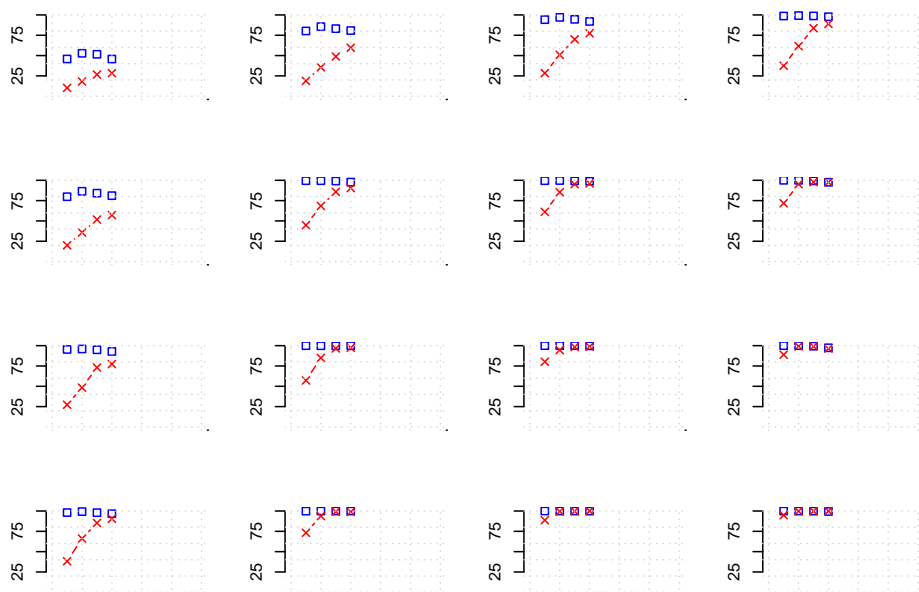


Figura 7.8: Efecto de 100 ms. sobre la cola

muestras de tamaño 7. Esto puede estructurarse mentalmente como 2 matrices (una por condición) de 32 filas (sujetos) y 7 columnas (items) por condición. La distribución *ExGauss* básica se basa en parámetros  $\mu = 400$ ,  $\sigma = 40$  y  $\tau = 200$ .

Se aplicarán diferentes efectos sobre los parámetros  $\mu$ ,  $\sigma$  y  $\tau$  de la distribución *ExGauss* original.

Además, en las diferentes muestras dispondremos de un máximo de 10 % de outliers.

Para cada variación de los parámetros se mostrarán y analizarán tres clases de gráficos, que pretenden analizar cada uno un aspecto diferente. Los tres tipos de gráficos (múltiples) se organizan en una presentación de  $4 \times 3$  subgráficos, cada fila analiza un tipo de transformación de los ya planteados y las columnas analizan los casos sin outliers y con un 10 % de outliers.

Los tres tipos de gráficos son:

- *QQ-Plot*. Comparación de los residuos con la recta de regresión lineal. Una única repetición del experimento.
- *Normalidad de residuos*. Histograma de residuos con una única repetición del experimento. La normalidad se analiza mediante el `_test` de *test de Pearson*  $\chi^2$ . El gráfico resultante serña rojo si no es normal y azul si el resultado del test anterior es positivo. El uso del test de Pearson se ha hecho mediante el uso de la librería de R *Nortest* (2015).
- *Histograma de residuos*. Como los dos primeros tipos de gráficos se basan en una única repetición del experimento correspondiente es necesario hacer un análisis estadístico basado en múltiples repeticiones. Tras 1000 repeticiones del experimento se muestra el histograma de las normas euclídeas de los vectores de residuos generados en cada repetición.

A continuación el código fuente de los diferentes tipos de gráficos.

```
outliers_glme_residuals_qq<-function(mu=400,sigma=40,tau=200,mu.ef=30,
                                     sigma.ef=0,tau.ef=0,p.outlier=0.1,nsujetos=32,nitems=7,metodo='
{
  # Generación de una muestra aleatoria

  A<-matrix(nrow=1,ncol=4);

  medias_base<-rnorm(n=nitems,mean=mu,sd=sigma)+rexp(n=1,rate=1/tau);
```

```

for(i in 1:nsujetos)
{
  A1<-matrix(nrow=nitems*2,ncol=4);
  muestra1<-rep(x=0,times=nitems);
  muestra2<-rep(x=0,times=nitems);

  for(j in 1:nitems)
  {
    # Condición 1
    muestra1[j]<-rnorm(n=1,mean=medias_base[j],sd=sigma)+
    rexp(n=1,rate=1/tau);

    # Condición 2
    muestra2[j]<-rnorm(n=1,mean=medias_base[j]+mu.ef,sd=sigma+sigma.ef)+
    rexp(n=1,rate=(1/tau)+(1/tau.ef));
  }

  aleatorio<-runif(n=nitems,min=0,max=1);
  muestra1[which(aleatorio<p.outlier)]<-
  muestra1[which(aleatorio<p.outlier)]+2000;

  aleatorio<-runif(n=nitems,min=0,max=1);
  muestra2[which(aleatorio<p.outlier)]<-
  muestra2[which(aleatorio<p.outlier)]+2000;

  if(metodo=="log")      # Transformación logaritmo
  {
    muestra1<-log(x=muestra1);
    muestra2<-log(x=muestra2);
  }

  if(metodo=="inv")      # Transformación inversa
  {
    muestra1<-1/muestra1;
    muestra2<-1/muestra2;
  }
}

```

```

    if(metodo=="inv2")      # Transformación inversa del cuadrado
    {
        muestra1<-1/(muestra1^2);
        muestra2<-1/(muestra2^2);
    }

    for (k in 1:nitems)
    {
        A1[k,]<-c(i,k,1,muestra1[k]);
        A1[k+nitems,]<-c(i,k,2,muestra2[k]);
    }

    A<-rbind(A,A1);
}

# La primera fila está vacía por ser parte de la creación de A y la elimino
A<-A[-1,];

# Organizamos los datos en un Data Frame para poder tratarlos mediante la
# función lmer de la librería lme4

datos<-data.frame(Subj=A[,1],Item=A[,2],SOA=A[,3],RT=A[,4]);

lme<-lmer(RT~SOA+(1+SOA|Item)+(1+SOA|Subj),data=datos,REML=F);

Res<-summary(lme)$residuals;  # residuos
qqnorm(Res,main="");
qqline(Res);
}

outliers_glme_residuals_qqhist<-function(mu=400,sigma=40,tau=200,mu.ef=30,
                                         sigma.ef=0,tau.ef=0,p.outlier=0.1,nsujetos=32,nitems=7,nr
metodo="nocut")
{
    v_histograma<-rep(x=0,times=nrep);
    for(nrepeticion in 1:nrep)

```

```

{
  # Generación de una muestra aleatoria

  A<-matrix(nrow=1,ncol=4);

  medias_base<-rnorm(n=nitems,mean=mu,sd=sigma)+rexp(n=1,rate=1/tau);

  for(i in 1:nsujetos)
  {
    A1<-matrix(nrow=nitems*2,ncol=4);
    muestra1<-rep(x=0,times=nitems);
    muestra2<-rep(x=0,times=nitems);

    for(j in 1:nitems)
    {
      # Condición 1
      muestra1[j]<-rnorm(n=1,mean=medias_base[j],sd=sigma)+
      rexp(n=1,rate=1/tau);

      # Condición 2
      muestra2[j]<-rnorm(n=1,mean=medias_base[j]+mu.ef,
      sd=sigma+sigma.ef)+rexp(n=1,rate=(1/tau)+(1/tau.ef));
    }

    aleatorio<-runif(n=nitems,min=0,max=1);
    muestra1[which(aleatorio<p.outlier)]<-
    muestra1[which(aleatorio<p.outlier)]+2000;

    aleatorio<-runif(n=nitems,min=0,max=1);
    muestra2[which(aleatorio<p.outlier)]<-
    muestra2[which(aleatorio<p.outlier)]+2000;

    if(metodo=="log") # Transformación logaritmo
    {
      muestra1<-log(x=muestra1);
      muestra2<-log(x=muestra2);
    }
  }
}

```



```

}

if(metodo=="inv")      # Transformación inversa
{
  muestra1<-1/muestra1;
  muestra2<-1/muestra2;
}

if(metodo=="inv2")     # Transformación inversa del cuadrado
{
  muestra1<-1/(muestra1^2);
  muestra2<-1/(muestra2^2);
}

for (k in 1:nitems)
{
  A1[k,]<-c(i,k,1,muestra1[k]);
  A1[k+nitems,]<-c(i,k,2,muestra2[k]);
}
A<-rbind(A,A1);
}

# La primera fila está vacía por ser parte de la creación de A y la elimino
A<-A[-1,];

# Organizamos los datos en un Data Frame para poder tratarlos mediante
# la función lmer de la librería lme4

datos<-data.frame(Subj=A[,1],Item=A[,2],SOA=A[,3],RT=A[,4]);

lme<-lmer(RT~SOA+(1+SOA|Item)+(1+SOA|Subj),data=datos,REML=F);

# Res<-summary(lme)$residuals; # residuos
# v_histograma[nrepeticion]<-sqrt(sum(Res^2));

Res<-summary(lme)$residuals; # residuos

```

```

x <- c(qnorm(.25), qnorm(.75));
y <- quantile(Res, c(.25, .75));
pdte <- diff(y)/diff(x);
int <- y[1] - pdte * x[1]; # 0 int <- y[2] - pdte * x[2]
diferencia=rep(x=0,times=nsujetos*nitems*2);
for(k in 1:nsujetos*nitems*2)
{
  # diferencia entre los residuos y la recta QQ
  diferencia[k]<-Res[k]-(int+pdte*Res[k]);
}
v_histograma[nrepeticion]<-sqrt(sum((Res-diferencia)^2));
}

return(v_histograma);
}

```

```

outliers_glme_residuals_hist<-function(mu=400,sigma=40,tau=200,mu.ef=30,
                                         sigma.ef=0,tau.ef=0,p.outlier=0.1,nsujetos=32,ni
{
  # Generación de una muestra aleatoria

  A<-matrix(nrow=1,ncol=4);

  medias_base<-rnorm(n=nitems,mean=mu,sd=sigma)+rexp(n=1,rate=1/tau);

  for(i in 1:nsujetos)
  {
    A1<-matrix(nrow=nitems*2,ncol=4);
    muestra1<-rep(x=0,times=nitems);
    muestra2<-rep(x=0,times=nitems);

    for(j in 1:nitems)
    {
      # Condición 1
      muestra1[j]<-rnorm(n=1,mean=medias_base[j],sd=sigma)+rexp(n=1,rate=1/tau);

      # Condición 2

```

```

    muestra2[j]<-rnorm(n=1,mean=medias_base[j]+mu.ef,sd=sigma+sigma.ef)+
    rexp(n=1,rate=(1/tau)+(1/tau.ef));
  }

  aleatorio<-runif(n=nitems,min=0,max=1);
  muestra1[which(aleatorio<p.outlier)]<-
  muestra1[which(aleatorio<p.outlier)]+2000;

  aleatorio<-runif(n=nitems,min=0,max=1);
  muestra2[which(aleatorio<p.outlier)]<-
  muestra2[which(aleatorio<p.outlier)]+2000;

  # Transformación logaritmo
  if(metodo=="log")
  {
    muestra1<-log(x=muestra1);
    muestra2<-log(x=muestra2);
  }

  # Transformación inversa
  if(metodo=="inv")
  {
    muestra1<-1/muestra1;
    muestra2<-1/muestra2;
  }

  # Transformación inversa del cuadrado
  if(metodo=="inv2")
  {
    muestra1<-1/(muestra1^2);
    muestra2<-1/(muestra2^2);
  }

  for (k in 1:nitems)
  {
    A1[k,]<-c(i,k,1,muestra1[k]);
  }

```

```

    A1[k+nitems,]<-c(i,k,2,muestra2[k]);
  }

  A<-rbind(A,A1);
}

# La primera fila está vacía por ser parte de la creación de A y la elimino
A<-A[-1,];

# Organizamos los datos en un Data Frame para poder tratarlos
# mediante la función lmer de la librería lme4

datos<-data.frame(Subj=A[,1],Item=A[,2],SOA=A[,3],RT=A[,4]);

lme<-lmer(RT~SOA+(1+SOA|Item)+(1+SOA|Subj),data=datos,REML=F);

Res<-summary(lme)$residuals; # residuos
pvalor<-pearson.test(Res)$p.value;

if (pvalor>=0.5)
{
  hist(Res,xlab="",ylab="",main="",border="blue",breaks=20);
}
if (pvalor<0.5)
{
  hist(Res,xlab="",ylab="",main="",border="red",breaks=20);
}
media<-mean(Res);
desviacion<-sd(Res);
abline(v=media,lty="dashed",col="green");
abline(v=media-desviacion,lty="dashed",col="yellow");
abline(v=media+desviacion,lty="dashed",col="yellow");
}

```

### 7.5.1 Sin efecto sobre ningún parámetro ni variabilidad entre sujetos

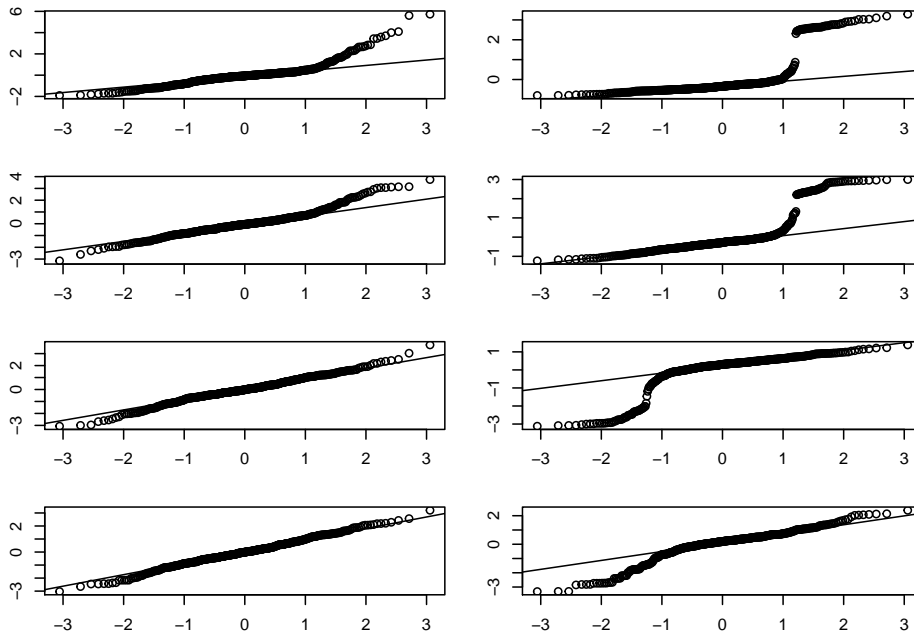


Figura 7.9: Sin efectos ni variabilidad intersujetos. QQ-Norm.

Sin efecto sobre ningún parámetro ni variabilidad entre sujetos se observa que, concretamente la transformación  $\frac{1}{t^2}$ , ofrece mejor adaptación de los residuos a la recta Cuantil-Cuantil, apreciable en los residuos de menor valor. En concreto, véase el caso con outliers.

En el caso sin outliers, al pasar de la transformación  $\frac{1}{TR}$  a  $\frac{1}{TR^2}$  se mantiene la simetría y centralidad. Analizando la diferencia entre los residuos y la recta normal correspondiente, particularmente en el caso con un 10 % de outliers, se observa sensible aumento de la media y disminución de la desviación.

### 7.5.2 Efecto sobre la media y sin variabilidad entre sujetos

Los residuos disminuyen al aplicar un efecto de 10 ms. sobre la media, concretamente en la transformación  $\frac{1}{T^2}$ .

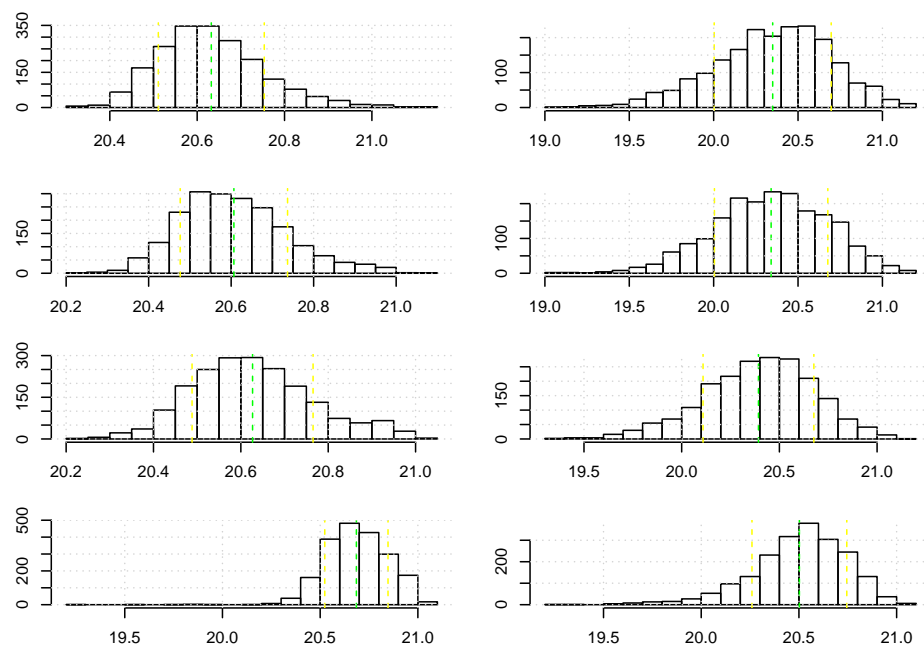


Figura 7.10: Sin efectos ni variabilidad intersujetos. Histograma de suma de residuos

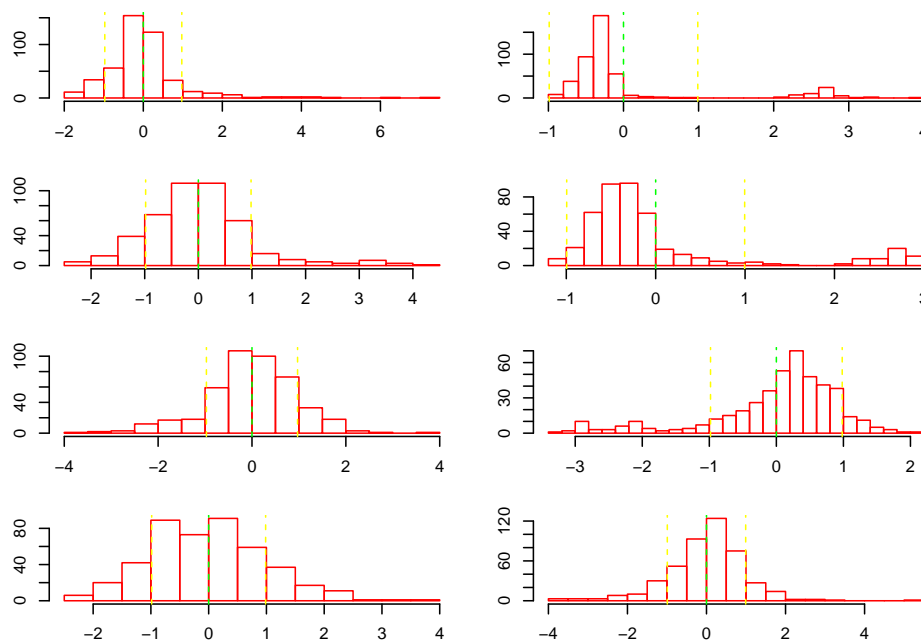


Figura 7.11: Sin efectos ni variabilidad intersujetos. Histograma.

Con un efecto de  $10\text{ ms.}$  sobre la media parece observarse una disminución de los residuos y mejor adaptación comparando con la distribución normal, particularmente en el caso de la transformación  $\frac{1}{T R^2}$  y aplicando un  $10\%$  de outliers. En concreto, la adaptación a la recta de regresión normal parece mejorar en los tiempos de respuesta bajos.

En un único experimento, la transformación  $\frac{1}{T R^2}$  parece disminuir el número de residuos alejados de la distribución normal, en contrecorreo los residuos del ala izquierda.

Al aumentar a  $20\text{ ms.}$  el efecto sobre la media no se observa especial diferencia en los casos sin outliers entre las dos últimas transformaciones. Sin embargo, al comparar estas dos mismas transformaciones en el respectivo caso  $10\%$  de outliers, la norma residual parece estar más concentrada en torno a los valores centrales. Esto parece corresponderse con la gráfica de comparación con la distribución normal, en que se aprecia entre los mismos casos mejor ajuste a la normal en los valores bajos. Ligera disminución de los residuos de valor más alto.

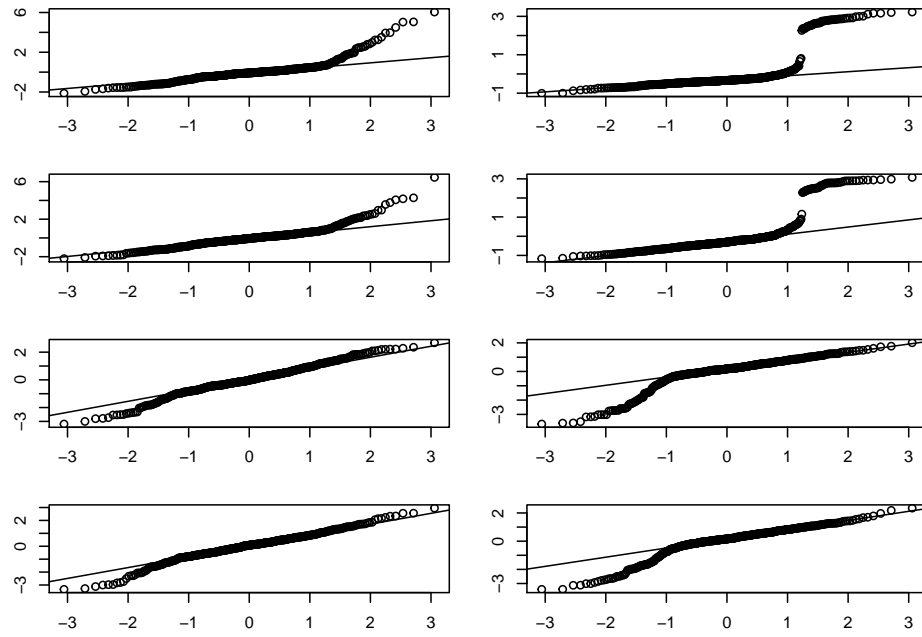


Figura 7.12: Efecto sobre la media 10 ms. QQ-Norm.

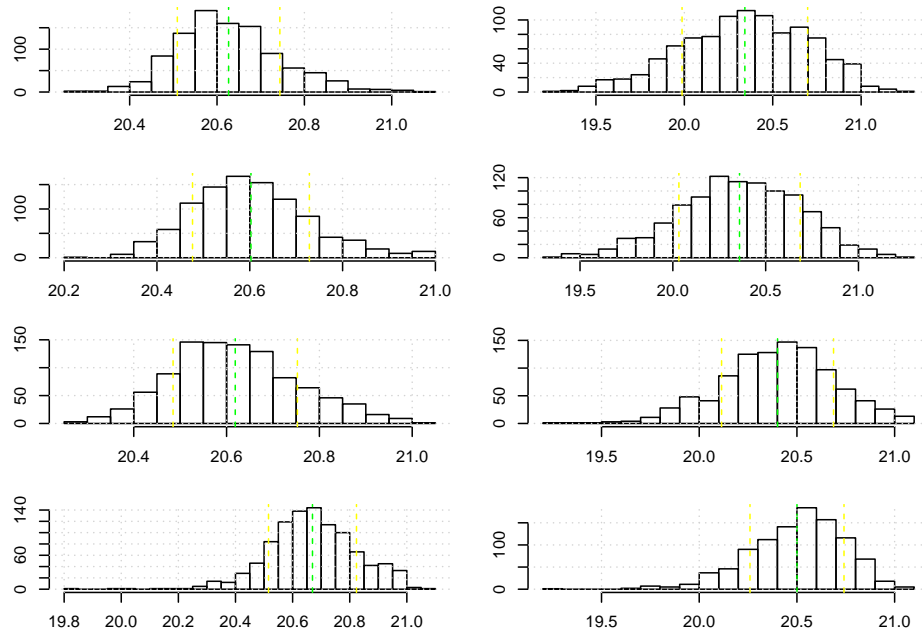


Figura 7.13: Efecto sobre la media 10 ms. Histograma de suma de residuos



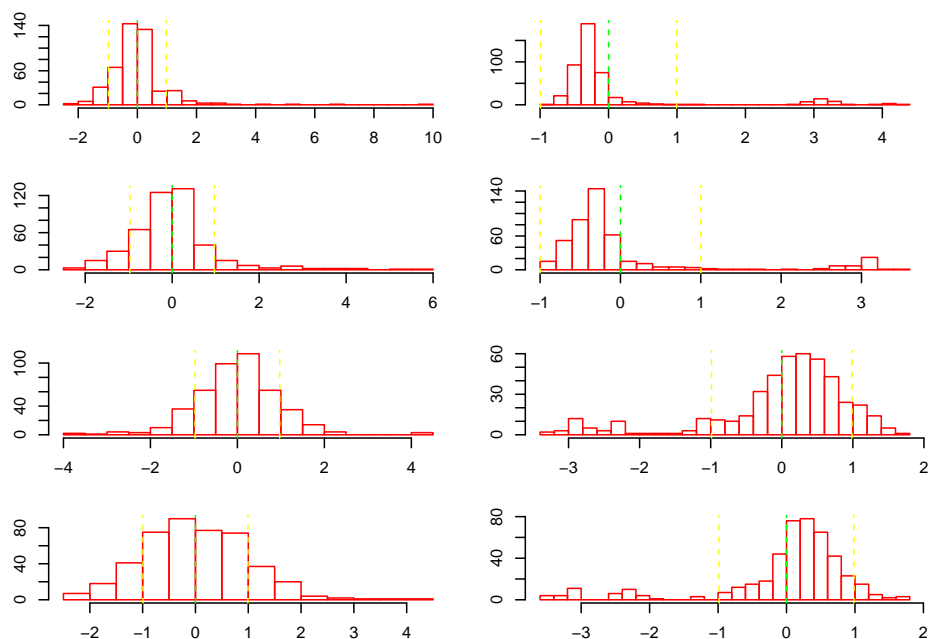


Figura 7.14: Sin efectos ni variabilidad intersujetos. Histograma.

Con efectos de  $40\text{ ms.}$  sobre la media parece mejorar sensiblemente el ajuste de los mismos a la distribución normal correspondiente.

Con 1000 repeticiones del experimento la distancia a la correspondiente distribución normal parece agruparse más en los valores centrales.

Finalmente, al aplicar un efecto de  $40\text{ ms.}$  sobre la media, la adaptación de los residuos a la recta cuantil-cuantil parece mejorar también, particularmente al comparar las dos últimas transformaciones. Además, el test de Pearson parece señalar normalidad en los residuos con la última transformación, en el caso sin outliers.

### 7.5.3 Efecto sobre la cola y sin variabilidad entre sujetos

Con efecto sobre  $\tau$  de  $20\text{ ms.}$  mejora el ajuste de los mismos a la distribución normal.

Es destacable como con un efecto de  $50\text{ ms.}$  se obtiene mayor centralización de los residuos y mejor ajuste a la distribución normal de los mismos.

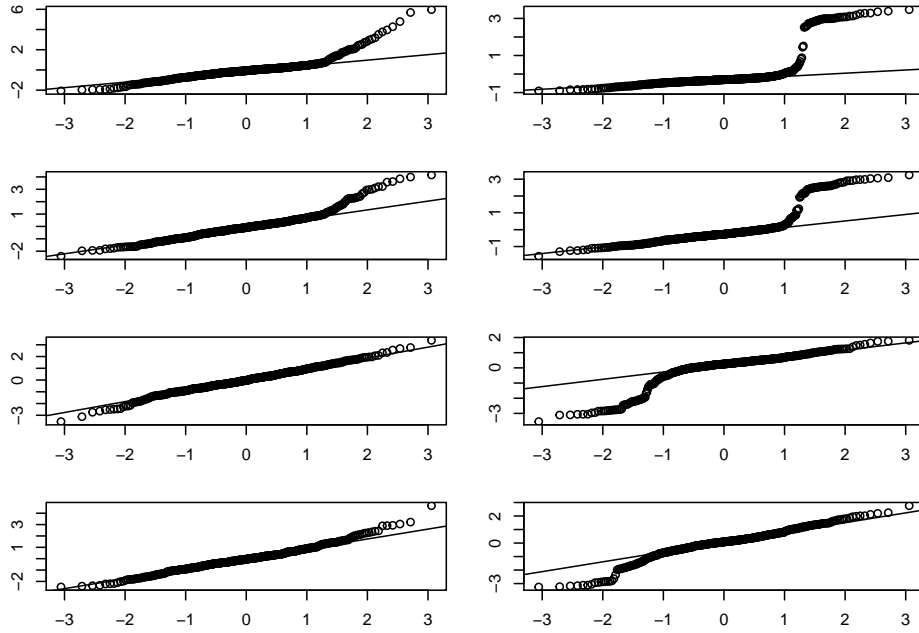


Figura 7.15: Efecto sobre la media 20 ms. QQ-Norm.

Con un efecto fijo, de hasta  $100\text{ ms.}$ , la transformación  $\frac{1}{TR^2}$  mejora el ajuste de los residuos a la recta Cuantil-Cuantil. Con una transformación fijada, aumentar el efecto sobre la cola reduce la diferencia entre los resícuos y la correspondiente recta Cuantil-Cuantil.

Frente a las demás situaciones analizadas en este apartado, aplicar un efecto grande sobre la cola normaliza la distribución de los residuos, siempre y cuando no se apliquen outliers. Frente al caso sin outliers, puede verse que al aplicar un  $10\%$  de outliers (gráficos columna derecha) que ninguna de las transformaciones ofrece normalidad ni simetría pero que los casos  $\frac{1}{t}$  y  $\frac{1}{t^2}$  parecen invertir la asimetría de la distribución.

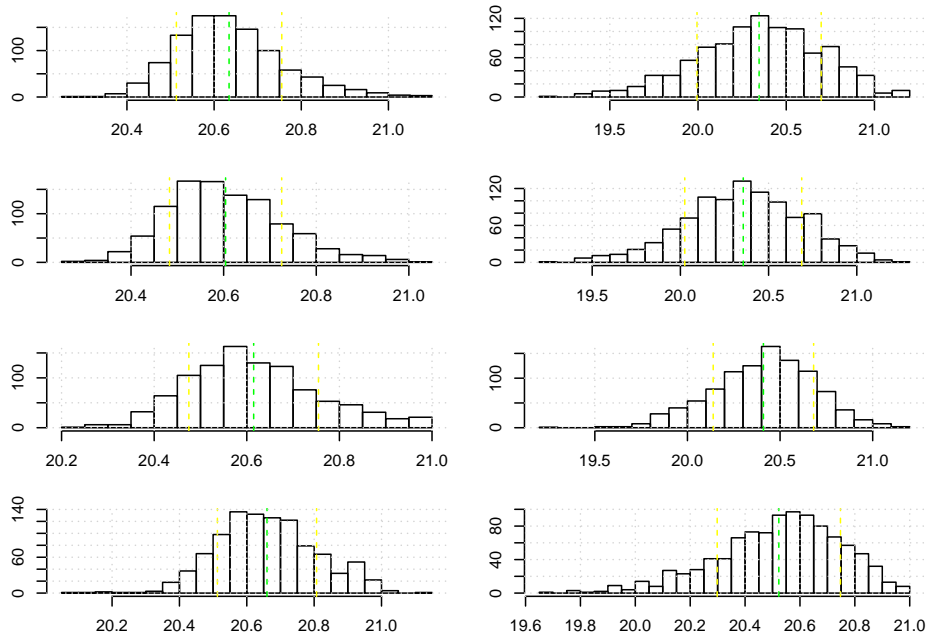


Figura 7.16: Efecto sobre la media 20 ms. QQ-Norm.

## 7.6 Conclusiones y posibles futuras vías de trabajo.

### 7.6.1 Análisis de potencia

El análisis de potencia se basará en número de sujetos e items creciente comparando particularmente las transformaciones  $\frac{1}{t}$  y  $\frac{1}{t^2}$ . Recordemos que la distribución base es  $ExGauss(\mu = 400, \sigma = 40, \tau = 200)$  (valores en *ms.*).

- Si no se aplica ningún tipo de efecto sobre los parámetros  $\mu$  y  $\tau$  puede verse que el error tipo I se mantiene entorno al 5 % con las diferentes transformaciones.
- Aplicando efecto sobre la media, la potencia entre los casos sin outliers y con un 10 % de outliers aumenta al cambiar de un método a otro, siendo su mayor valor en la transformación  $\frac{1}{t^2}$ . Sin embargo, con cada transformacion también disminuye la diferencia entre el caso sin outliers y con un 10 % de outliers. Los efectos grandes sobre la media hacen que

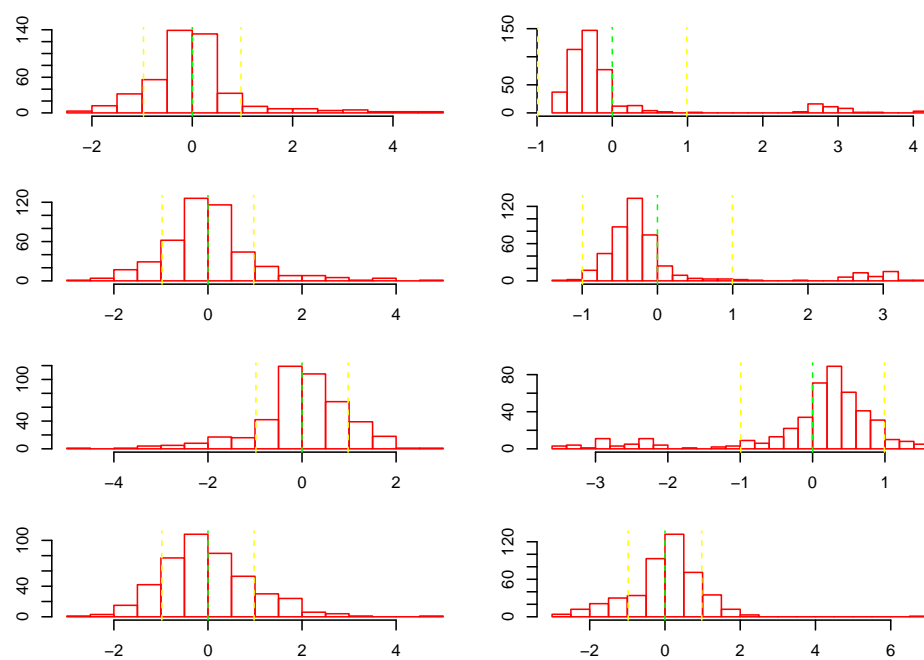


Figura 7.17: Efecto sobre la media 20 ms. Histograma.

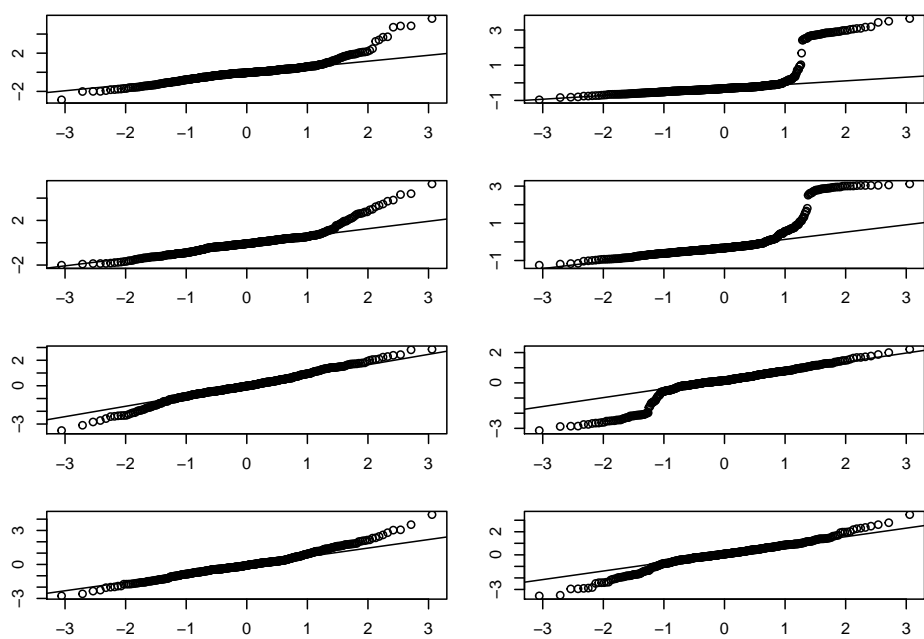


Figura 7.18: Efecto sobre la media 40 ms. QQ-Norm.

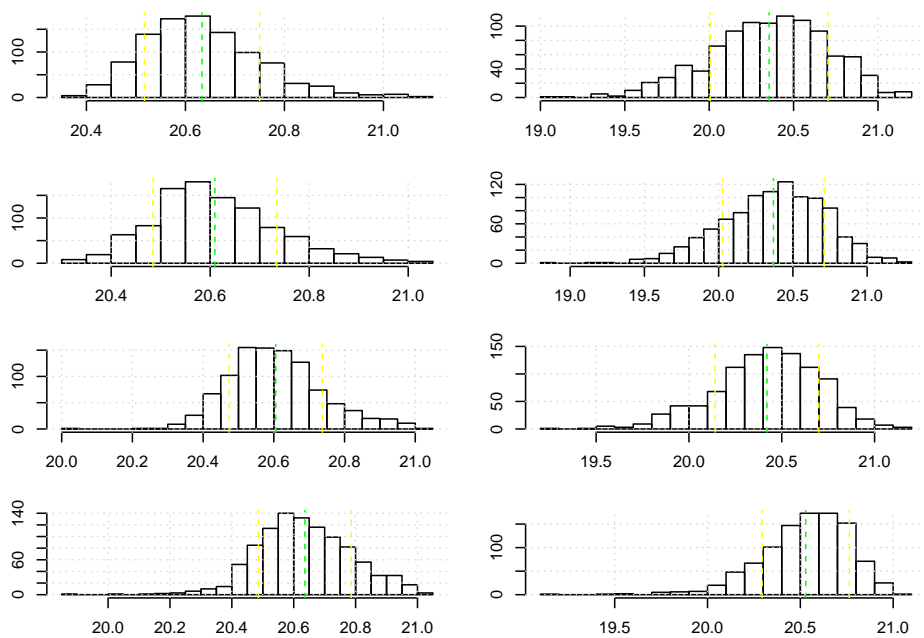


Figura 7.19: Efecto sobre la media 40 ms. Histograma de suma de residuos

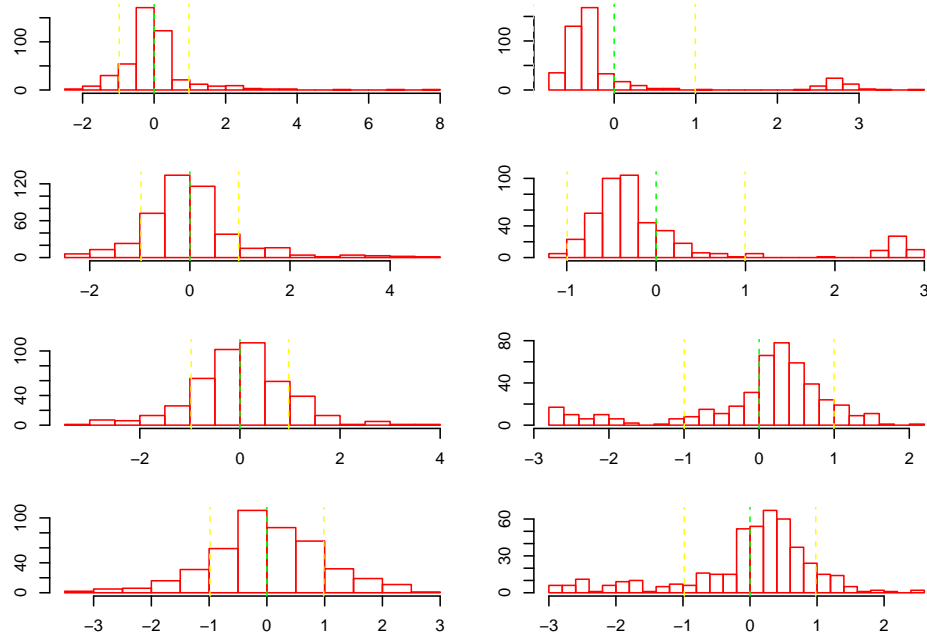


Figura 7.20: Efecto sobre la media 40 ms. Histograma.

se reduzca rápidamente la diferencia entre los casos sin outliers y con un 10 % de outliers.

- Con efectos sobre  $\tau$ , se observa que con cada transformación el caso con un 10 % de outliers se mantiene constante pero aumenta la potencia del caso sin outliers. De este modo, se reduce la diferencia de potencia entre los casos sin outliers y con un 10 % de outliers.
- Si se aplican diferentes efectos sobre los parámetros  $\mu$  y  $\tau$  puede observarse que al aumentar el efecto sobre alguno de dichos parámetros la diferencia entre las situaciones sin y con outliers disminuye considerablemente.

### 7.6.2 Análisis de residuos

Para el análisis de residuos el desarrollo se basa en dos condiciones. La primera condición está basada en una distribución  $ExGauss(\mu = 400, \sigma = 40, \tau = 200)$  aplicada sobre 32 sujetos y 7 items por sujeto. La segunda condición, con el mismo número de sujetos e items por sujeto, se basa en la misma distribución sobre la que se han aplicado diferentes efectos sobre  $\mu$  o  $\tau$ .

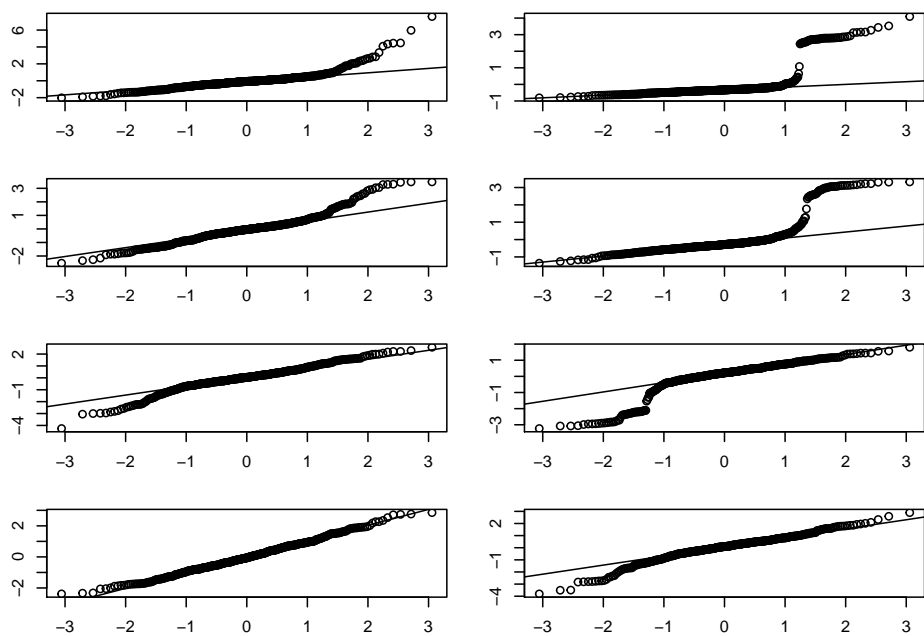


Figura 7.21: Efecto sobre la cola 20 ms. QQ-Norm.

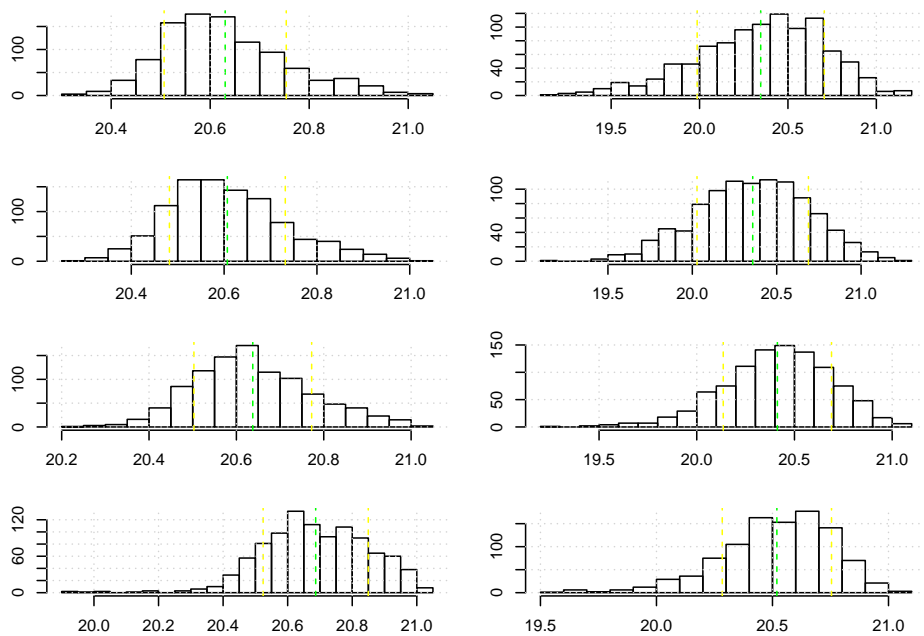


Figura 7.22: Efecto sobre la cola 20 ms. Histograma de suma de residuos.

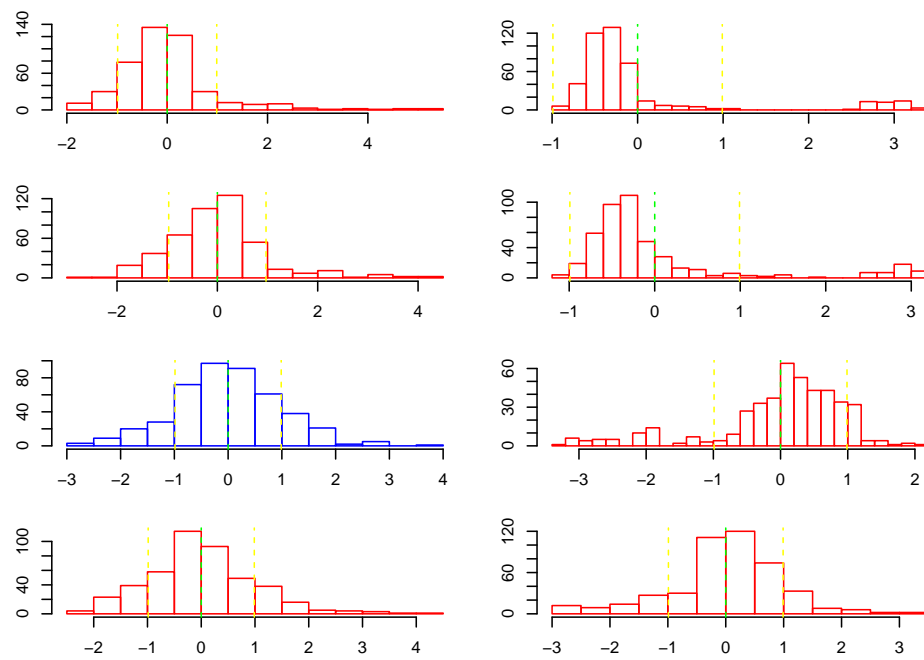


Figura 7.23: Efecto sobre la cola 20 ms. Histograma.



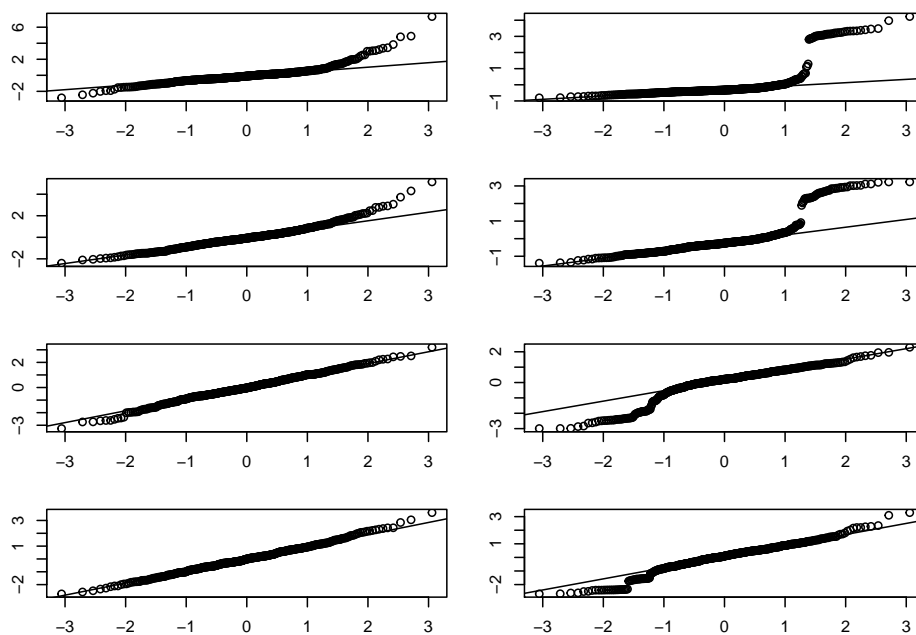


Figura 7.24: Efecto sobre la cola 50 ms. QQ-Norm.

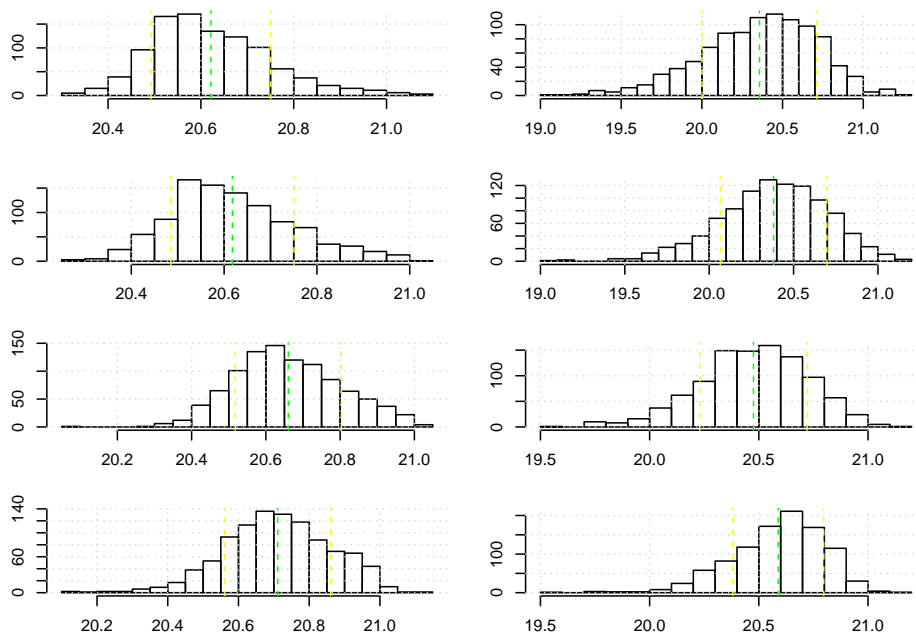


Figura 7.25: Efecto sobre la cola 50 ms. Histograma de suma de residuos.

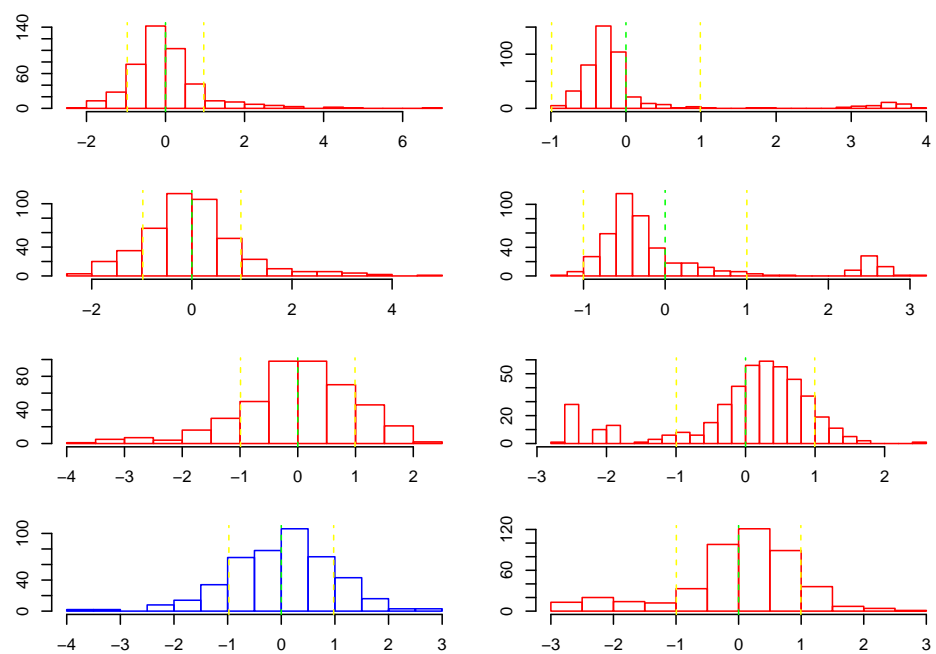


Figura 7.26: Efecto sobre la cola 50 ms. Histograma.

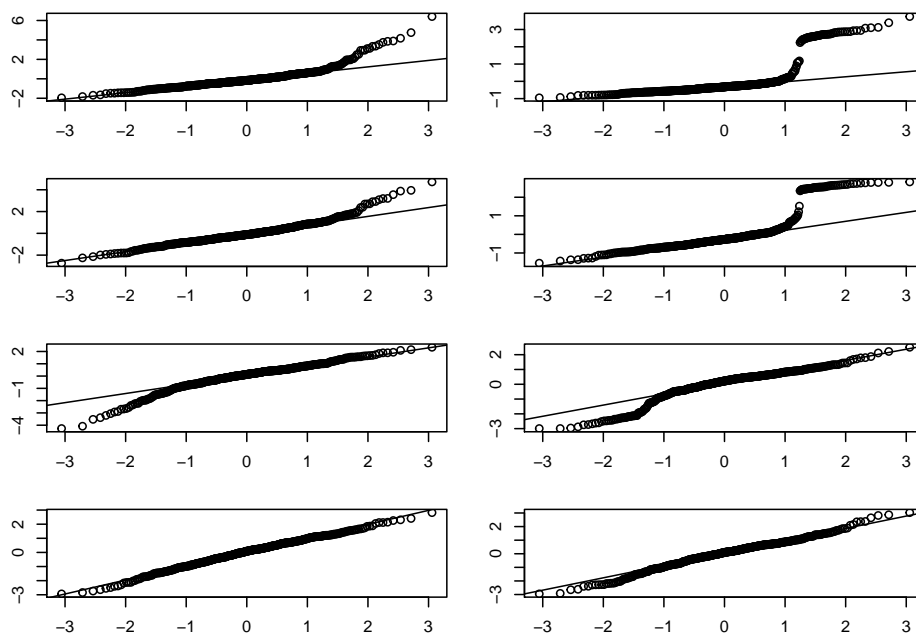


Figura 7.27: Efecto sobre la cola 100 ms. QQ-Norm.

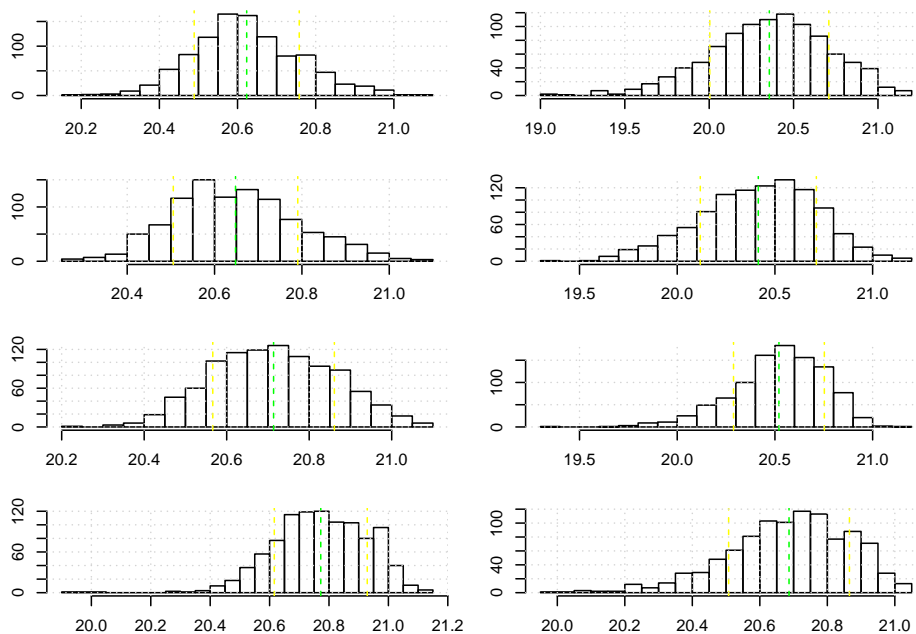


Figura 7.28: Efecto sobre la cola 100 ms. Histograma de suma de residuos.

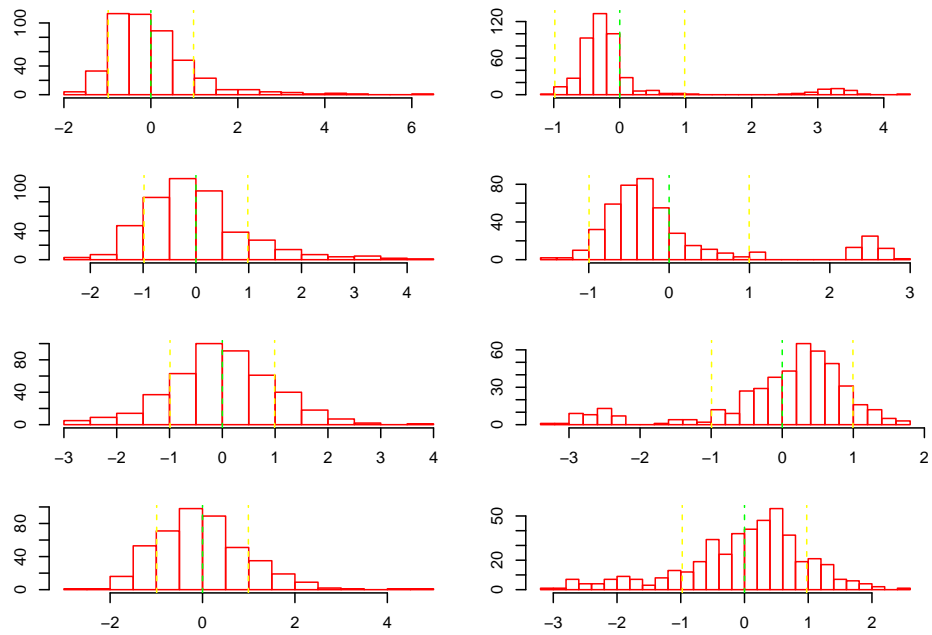


Figura 7.29: Efecto sobre la cola 100 ms. Histograma.

- Un incremento del efecto sobre  $\mu$  reduce la diferencia de los residuos a la correspondiente distribución normal.
- Aplicando efecto sobre la cola se observa mejor ajuste a la distribución normal.
- El histograma de residuos no se ajusta a una distribución normal, acorde al trabajo de Gellman et al. (2007).

## Parte IV

# Conclusiones



## Capítulo 8

# Resumen métodos y conclusiones

### 8.1 Presentación

Se muestra en este capítulo final un resumen de las conclusiones obtenidas durante el desarrollo del trabajo previo en sus diferentes bloques de desarrollo.

En todos los casos se trabaja sobre datos simulados computacionalmente y basados en una distribución *Ex-Gauss* para simular Tiempos de Respuesta en reconocimiento de palabras, como corresponde al ámbito de la Psicología Cognitiva y Neurolingüística.

### 8.2 Bloque I

Esta parte del trabajo es un acercamiento a los métodos de investigación en el área que nos ocupa y desde una perspectiva estadística y computacional.

## 8.2.1 Survival

### 8.2.1.1 Objetivo

Evaluar la potencia y posibles mejoras de una propuesta de método para calcular el punto de divergencia entre dos curvas correspondientes a dos condiciones diferenciadas.

### 8.2.1.2 Inicialización

En cuanto a los datos de partida, se generan por simulación computacional en base a una distribución  $ExGauss(\mu, \sigma, \tau)$ , o lo que es lo mismo, suma de una distribución Normal o Gauss  $Gauss(\mu, \sigma)$  y una exponencial  $Ex(\tau)$ . Sean en este caso  $\mu$  (media),  $\sigma$  (desviación típica) y  $\tau$  (cola). Con el conjunto base de datos anterior se generan por *bootstrapping* o remuestreo con reemplazamiento diferentes muestras según la estructura que se aclara en el desarrollo del método.

### 8.2.1.3 Método

Partimos de  $n_S$  sujetos sometidos a 2 condiciones y  $n_I$  items por condición. Consideramos las lecturas de cada condición por separado. Esto nos permite generar para cada condición  $n_S$  curvas de supervivencia. Se calcula finalmente la curva de supervivencia promedio de las  $n_S$  curvas anteriores. El objetivo último es determinar dos curvas de supervivencia (promedio), una por cada condición, que nos permiten extraer el punto de divergencia de ambas curvas, es decir, el punto de divergencia entre ambas condiciones.

### 8.2.1.4 Resultados

Centramos en este momento nuestra atención sobre un supersujeto sometido a dos condiciones, es decir, un sujeto con gran número de lecturas / items por condición.

Al aumentar el número de items, se reduce la mediana del punto de divergencia y los datos se centralizan más. Esto podría ser un indicio de la efectividad del método. Al compararse casos con poca y mucha asimetría positiva (recordemos la representación gráfica de la distribución *Ex-Gaussiana* empleada para



los tiempos de respuesta simulados) se observa mayor ritmo de decrecimiento del punto de divergencia en el caso con menor asimetría positiva.

Al aplicarse diferentes efectos sobre la media o la cola de la distribución, en ambos casos se reduce la media de los puntos de divergencia a la par que en el correspondiente diagrama de cajas aumenta el número de valores extremos.

Es en este momento, cuando aplicando la misma clase de efecto sobre la media ( $\mu$ ) o la cola ( $\tau$ ) vemos que no hay estabilidad en el cálculo del punto de divergencia al cambiar el efecto.

Los propios Reingold & Sheridan (2018) reconocen posteriormente a Reingold et al. (2012) la limitación del método propuesto a condiciones de base específicas.

Finalmente el método propuesto en el trabajo teórico original no es consistente. En el artículo posteriormente publicado Gomez et. al (2021) se rebaten de manera formal las bases del trabajo original de Reingold et al. (2012)

## **8.2.2 Redes neuronales**

### **8.2.2.1 Objetivo**

Se utiliza un modelo de redes autoorganizadas como una propuesta de emulación del proceso aprendizaje lector. En este modelo competitivo, ante la presentación de un estímulo, se determina la neurona más cercana en peso al estímulo y se ajustan las demás para parecerse en la mayor medida posible al estímulo. Se pretende que representaciones diferentes del mismo concepto lleguen a generar el mismo patrón de activación neuronal.

#### **8.2.2.2 Inicialización**

- Una mapa neuronal  $m \times n \times k$ , visto como  $m \times n$  neuronas entendidas como vectores  $k$ -dimensionales.
- Un alfabeto o conjunto de estímulos (letras) entendido como un conjunto de vectores  $k$ -dimensionales, inicialmente no relacionados con los vectores / neuronas del mapa neuronal inicial.

Aunque los estímulos-letras pueden corresponder al concepto minúscula o mayúscula, su representación inicial no es necesariamente similar. Es decir, existen

diferentes grados de similitud inicial entre los pares minúscula - mayúscula (“a” vs. “A”, “b” vs. “B”, “c” vs. “C”). Además, debe tenerse en cuenta que si en la representación inicial se presenta un componente fonético esto cambiaría el ritmo de aprendizaje posterior. Ambas cuestiones justifican los diferentes modelos experimentales diferenciados más adelante.

### 8.2.2.3 Método

#### 8.2.2.3.1 Algoritmo base

- Se toma aleatoriamente uno de los estímulos y se determina cuál de las neuronas del mapa es la más cercana en norma euclídea al estímulo presentado. Esta se conocerá como neurona activa.
- Las neuronas diferentes a la neurona activa ajustan su peso en base a su grado de similitud con el estímulo y su distancia a la neurona activa.
- Los dos primeros puntos se repiten un número predeterminado de veces  $N$ .

#### 8.2.2.3.2 Experimentos

- **Experimento 1.** El conjunto de estímulos es generado de forma aleatoria. Se han empleado vectores peso 5-dimensionales. Esta primera opción de generación de datos tiene en su contra que conceptos con una representación visual similar puedan representarse por vectores peso totalmente diferentes; y viceversa.
- **Experimento 2.** El conjunto de estímulos generado mediante lectura del mapa de bits de la imagen de la letra. Aquí se generan letras gráficamente, en formato mapa de bits, y se convierten en una lectura 0-1 dependiendo que cada unidad de la representación esté ocupada o no. Aquí los vectores estímulo tienen mayor longitud y el proceso de entrenamiento del cerebro se prevee más lento pero el objetivo es también una mayor rapidez de aprendizaje por ser los estímulos más cercanos a la realidad.
- **Experimento 3.** Se incorpora un componente fonético a la generación de los estímulos iniciales de forma que una parte del vector-estímulo coincide en los pares minúscula-mayúscula. Además de las ventajas del **Experimento 2** tengamos en cuenta que estímulos gráficamente diferentes, en los pares minúscula-mayúscula, tendrán una parte común

debido al componente fonético incorporado.

Por lo indicado, cada experimento se prevee ofrecerá un aprendizaje: \* cualitativamente más rápido y preciso; \* cuantitativamente (computacionalmente) más exigente, no tanto por el peso de los vectores estímulo como por la forma de generarlos. En **Experimento 2** y **Experimento 3** los estímulos no se generan de forma estrictamente aleatoria sino mediante un proceso de lectura de imagen (**Experimento 2**) o fijando una parte fonética del estímulo (**Experimento 3**).

#### 8.2.2.4 Resultados

Los pares minúscula-mayúscula generan, tras el proceso de entrenamiento / aprendizaje, el mismo patrón de activación neuronal o *ALI* (*Abstract Lineal Identity*), independientemente del grado de similitud visual entre la minúscula y la mayúscula.

En el caso del **Experimento 1**, al generarse los pesos de los estímulos de forma aleatoria, no se observa diferencia entre el resultado en un caso de similitud nula y uno de similitud moderada o alta.

En los casos del **Experimento 2** y **Experimento 3**, se tiene en cuenta tanto el grado de similitud visual como un componente fonético, lo que hace que en una similitud visual menor se aprecie al final del proceso un patrón de activación neuronal ligeramente diferente en los pares minúscula-mayúscula, frente a un alto grado de similitud de dicho patrón en un caso de alta similitud visual.

En resumen, en los sucesivos experimentos, se observa que el empleo de un componente visual y/o fonético genera un mejor patrón de activación neuronal en el par minúscula-mayúscula.

En el caso de aprendizaje de letras es claro que el grado de similitud del mapa neuronal de activaciones es mejor para la misma pareja minúscula / mayúscula en los sucesivos experimentos. En el caso de los mapas entrenados para el aprendizaje de palabras, al determinarse los pesos respectivos a partir del mapa entrenado de letras, el resultado es similar en unos y otros experimentos. Es decir, el entrenamiento previo de letras facilita el entrenamiento de palabras. Dicho de otro modo, un cerebro entrenado en reconocimiento de letras facilita el entrenamiento de palabras, aun cuando la diferencia del número de repeticiones para el entrenamiento sea del orden 1500 frente a 150 repeticiones.

Estos resultados nos hicieron plantearnos la progresión del aprendizaje, mediante el uso del Error de Precisión de la Proyección. Dicho error, aun siendo inferior en el caso del Experimento 1, se estabiliza notablemente más rápido en los casos de Experimento 2 y Experimento 3.

## 8.3 Bloque II

### 8.3.1 Análisis de outliers

#### 8.3.1.1 Objetivo

Dentro de un conjunto de datos se entiende como datos extremos o *outliers* aquellos valores inconsistentes con el resto de valores. Para evitar que esta clase de datos afecten a las conclusiones estadísticas es frecuente el empleo de diferentes transformaciones como  $TR$  (No cut / sin transformación),  $\log(TR)$  o  $\frac{1}{TR}$ . El estudio de robustez es un resultado conocido en base a un análisis lineal y uso del **t-test**.

El objetivo de esta sección es doble:

1. Introducir como transformación alternativa  $\frac{1}{TR^2}$  y analizarla igualmente mediante un modelo lineal y **t-test**.
2. Análisis de las diferentes transformaciones, incluida la nueva propuesta  $\frac{1}{TR^2}$ , mediante el uso de *Modelo Lineal Generalizado Mixto*.

#### 8.3.1.2 Inicialización

Los datos se generan en base a una distribución base  $ExGauss(\mu, \sigma, \tau)$ . Se confrontan los datos a partir de dicha distribución con aquellos datos generados con un 10 % de *outliers*.

#### 8.3.1.3 Método

Se analizan diferentes transformaciones sobre los datos para evaluar la bondad de la transformación correspondiente tanto desde el punto de vista de los modelos lineales como de los modelos lineales generalizados mixtos.

#### 8.3.1.4 Resultados

##### 8.3.1.4.1 Análisis de potencia

###### 8.3.1.4.1.1 Error Tipo I

En este apartado no se aplica efecto sobre ninguno de los parámetros de la distribución de  $TR$ .

Tanto en el caso del *Modelo Lineal* como del *Modelo Lineal Generalizado Mixto* el Error Tipo I se mantienen por debajo del 5 %.

###### 8.3.1.4.1.2 Efecto sobre la media

Se aplican efectos crecientes sobre la media de 20, 50 y 100 ms.

La diferencia entre el caso *sin outliers* y con un 10 % de *outliers* se reduce al pasar por cada transformación *No cut*,  $\log(TR)$ ,  $\frac{1}{TR}$  y  $\frac{1}{TR^2}$ . Dicha diferencia, notable con un efecto pequeño para las primeras transformaciones, se ve casi anulada para estas mismas transformaciones conforme se aumenta el efecto sobre la media.

###### 8.3.1.4.1.3 Efecto sobre la cola

Se aplican efectos crecientes sobre la cola de 20, 50 y 100 ms.

De forma análoga a lo que ocurre al aplicar los efectos sobre la media, la diferencia entre los casos sin outliers y con un 10 % de outliers se reduce con cada transformación y también conforme aumenta el tamaño del efecto.

##### 8.3.1.4.2 Análisis de residuos

###### 8.3.1.4.2.1 Efecto sobre la media

Independientemente del efecto aplicado sobre la media la transformación del tipo  $\frac{1}{TR^2}$  ofrece mejores resultados que las anteriores en cuanto a que mejora la aproximación de los residuos a una distribución normal.

#### **8.3.1.4.2.2 Efecto sobre la cola**

Los efectos crecientes sobre la cola mejoran la aproximación de los residuos a una distribución normal.

## Capítulo 9

# Recapitulación de conclusiones y posibles vías futuras de trabajo

### 9.1 Análisis de Supervivencia

En lo referente al apartado de Análisis de Supervivencia, aunque la técnica se planteó como novedosa, los resultados extraídos de las simulaciones computacionales pertinentes demostraron la limitada efectividad de la misma. Hubiera sido desable obtener resultados más optimistas para incorporar esta técnica de cálculo del punto de divergencia a la batería de opciones actualmente disponibles a tal efecto.

### 9.2 Redes Neuronales Autoorganizadas

Esta es una técnica base de contrastado uso en diferentes áreas de trabajo. Sin embargo, la falta de retroalimentación por parte del autor original y de la publicación, mostraron la conveniencia de utilizar esta sección como un acercamiento a las técnicas computacionales en neurociencia cognitiva y no como el bloque principal de trabajo. No obstante, los resultados fueron similares a los

del trabajo de base. Contando con el contraste de la información con el trabajo original, podría haberse sometido a esta técnica a diferentes tests de estrés frente a datos reales para contrastar el grado de acercamiento al aprendizaje real.

Tanto en el caso de letras como de palabras, el resultado del mapa entrenado final, muestra clara dominancia de uno de los estímulos en la representación gráfica.

A modo de posible vía de trabajo futura en este punto sería interesante determinar cuál puede ser el motivo o motivos de esta situación: \* La forma de generar los pesos de los estímulos, \* el número de repeticiones, \* la influencia de los últimos estímulos presentados para el entrenamiento del mapa, \* influencia del orden de presentación de los estímulos, \* tamaño del mapa neuronal final, \* etc.

### 9.3 Tratamiento de outliers

Llegamos a la sección final, dedicada al estudio de la transformación  $\frac{1}{T^2}$  frente a las previas  $T$ ,  $\log(T)$  y  $\frac{1}{T}$ , tanto mediante el *Modelo Lineal* como mediante el uso del más reciente *Modelo Lineal Generalizado Mixto*.

En ambos tipos de modelización, el Error Tipo I se mantiene por debajo del 5 % en el caso en que no hay efecto sobre ningún parámetro de la distribución base  $Ex - Gauss(\mu, \sigma, \tau)$ . Cabe señalar aquí, que al aumentar el número de sujetos y/o items, la diferencia entre los casos sin outliers y con un 10 % de outliers se reduce notablemente con la transformación  $\frac{1}{T^2}$ .

La transformación  $\frac{1}{T^2}$  también hace que los residuos se aproximen en mayor grado a una distribución Normal si los comparamos con las anteriores transformaciones.

En esta parte del trabajo, que es finalmente el núcleo del mismo, sería deseable someter al mismo tipo de tests, en lugar de sobre datos simulados computacionalmente, sobre datos procedentes de lecturas sobre sujetos reales.

En lo referente al uso de la transformación  $\frac{1}{T^2}$ , ya se ha analizado como la diferencia entre los casos sin outliers y con un 10 % de outliers se reduce con esta transformación frente a las anteriores, ya conocidas. Esta diferencia se hace casi



nula, lo que nos lleva a pensar que no tendría sentido estudiar transformaciones del tipo  $\frac{1}{T^n}, n > 2$ .



**Parte V**

**Final**



## Capítulo 10

# Agradecimientos

El desarrollo que sigue ha sido dirigido por el Profesor Manuel Perea Lara de la *Universitat de València*, al cuál agradezco su dirección académica, apoyo y orientación.

Han sido fundamentales también las codirecciones de los Profesores Pablo Gómez, de California State University San Bernardino, Palm Desert (California, EEUU), y Ana Marcet Herranz, de la Universitat de València. Al primero corresponde el asesoramiento en lo concerniente a la parte computacional del trabajo; y a la segunda las mejoras en la redacción y organización del documento.

No quisiera dejar de mencionar a la Dra. Eva Rosa de la *Univerditat de València* que me puso en contacto con quien acabó siendo mi director de Tesis, permitiéndome orientar mi preparación teórica de base en matemáticas hacia un área más aplicada.

Reciba también una mención el *Centro de Cálculo* de la *Universitat de València*, por los medios técnicos aportados y particularmente el soporte al uso de los mismos, que permitieron desarrollar la amplia batería de simulaciones computacionales.

Sin la colaboración de todos ellos no habría sido posible completar satisfactoriamente este proyecto.



# Bibliografía

Baayen, R., Davidson, D., & Bates, D. (2008). Mixed-effects modeling with crossed random effects for subjects and items. *Journal of Memory and Language*, 59(4), 390-412.

Balota, D., & Spieler, D. (1999). Word frequency, repetition, and lexicality effects in word recognition tasks: Beyond measures of central tendency. *Journal of Experimental Psychology: General*, 128, 32-55.

Barnett, V., & Lewis, T. (1994). *Outliers in Statical Data*. New York: John Wiley & Sons.

Barra, D. J., Levyb, R., Scheepersa, C., & Tilyc, H. J. (2013). Random effects structure for confirmatory hypothesis testing: Keep it maximal. *Journal of memory and language*, 68, 255-278.

Bates, D., Mächler, M., Bolker, B., Bates, D., Mächler, M., & and, B. B. (2015). Fitting Linear Mixed-Effects Models Using lme4. *Journal of Statistical Software*, 67.

Boles, D. B., & Clifford, J. E. (1989). An upper- and lowercase alphabetic similarity matrix, with derived generation similarity values. *Behavior Research Methods, Instruments, & Computers*, 21(6), 579-586.

Bolker, B. M., Brooks, M. E., Clark, C. J., Geange, S. W., Poulsen, J. R., Stevens, M. H. H., & White, J.-S. S. (2009). Generalized linear mixed models: a practical guide for ecology and evolution. *Trends in Ecology & Evolution*, 3(24), 127-135.

Efron, B. (1979). Bootstrap methods: Another look at the jackknife. *The*

*Annals of Statistics*, 7(1), 1-26.

Efron, B. (1987). Better Bootstrap Confidence Intervals. *Journal of the American Statistical Association*, 82(397), 171-185.

Freeberg, T., & Lucas, J. (2009). Pseudoreplication is (still) a problem. *Journal of Comparative Psychology*, 123(4), 450-451.

Gelman, A., & Hill, J. (2007). *Data analysis using regression and multilevel/hierarchical models*. Cambridge: Cambridge University Press.

Gómez, P., Breithaupt, J., Perea, M., & Rouder, J. N. (2021). Are divergence point analyses suitable for response time data? *Behavior Research Methods*, 53(1), 49-58.

Gross, J., & Ligges, U. (2015). *Package «nortest»*.

Heathcote, A., Popiel, S., & Mewhort, D. (1991). Analysis of response time distributions: An example using the Stroop task. *Psychological Bulletin*, 109(2), 340-347.

Kangas, J., Kohonen, T., & Laaksonen, J. (1990). Variants of self-organizing maps. *IEEE transactions on neural networks*, 1(1), 93-99.

Knuth, D. E. (1984). *The TeXbook, Computers and Typesetting*. Addison-Wesley.

Kohonen, T. (1990). The Self-Organizing Map. *Proceedings of the IEEE*, 78(9), 1464-1480.

Lazic, S. E. (2010). The problem of pseudoreplication in neuroscientific studies: is it affecting your analysis? *BMC Neuroscience*, 11, 1-17.

Lerner, I., Armstrong, B. C., & Frost, R. (2014). What can we learn from learning models about sensitivity to letter-order in visual word recognition? *Journal of Memory and Language*, 77, 40-58.

Luck, S. J. (2005). *An Introduction to the Event-Related Potential Technique*. The MIT Press.

Luck, S. J., & Kappenman, E. (2012). *The Oxford Handbook of Event-Related Potential Components*. Oxford University Press.

Nelder, J., & Wedderburn, W. (1972). Generalized linear models. *Journal of*



*the Royal Statistical Society-Series A*, 135(3), 370-384.

Perea, M., Jiménez, M., Talero, F., & López-Cañada, S. (2014). Letter-case information and the identification of brand names. *British Journal of Psychology*, 106(1), 162-173.

Perea, M., Rosa, E., & Marcet, A. (2017). Where is the locus of the lowercase advantage during sentence reading? *Acta Psychologica*, 177, 30-35.

Pinheiro, J., & Bates, D. (2000). *Mixed-Effects Models in S and SPLUS*. New York: Springer.

Polk, T. A., & Farah, M. J. (1997). A Simple Common Contexts Explanation for the Development of Abstract Letter Identities. *Neural Computation*, 9, 1277-1289.

Polk, T. A., Lacey, H. P., Nelson, J. K., Demiralp, E., Newman, L. I., Krauss, D. A., ... Farah, M. J. (2009). The development of abstract letter representations for reading: Evidence for the role of context. *Cognitive Neuropsychology*, 26(1), 70-79.

Polk, T. A., Stallcup, M., Aguirre, G. K., Alsop, D. C., D'Esposito, M., Dentre, J. A., & Farah, M. J. (2002). Neural Specialization for Letter Recognition. *Journal of Cognitive Neuroscience*, 14(2), 145-159.

Ratcliff, R. (1979). Group reaction time distributions and an analysis of distribution statistics. *Psychological Bulletin*, 86(3), 446-461.

Ratcliff, R. (1993). Methods for Dealing With Reaction Time Outliers. *Psychological Bulletin*, 114(3), 503-532.

Ratcliff, R., & Murdock, B. B. (1976). Retrieval processes in recognition memory. *Psychological Review*, 83(3), 190-214.

Reingold, E. M., & Sheridan, H. (2018). On using distributional analysis techniques for determining the onset of the influence of experimental variables. *Quarterly Journal of Experimental Psychology*, 71(1), 260-271.

Reingold, E. M., Reichle, E. D., Glaholt, M. G., & Sheridan, H. (2012). Direct lexical control of eye movements in reading: Evidence from a survival analysis of fixation durations. *Cognitive Psychology*, 65(2), 177-206.

Schielzeth, H., & Forstmeier, W. (2009). Conclusions beyond support: over-

confident estimates in mixed models. *Behavioral Ecology*, 20(2), 416-420.

Team, T. R. (2015). *RStudio: Integrated Development Environment for R*. Boston, MA: RStudio, Inc.

Team, T. R. C. (2006). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. Recuperado a partir de <https://www.R-project.org/>

Vergara-Martínez, M., Perea, M., Gómez, P., & Swaab, T. Y. (2013). ERP correlates of letter identity and letter position are modulated by lexical frequency. *Brain and Language*, 125(1), 11-27.

WIKI del Servicio de Informática SiUV. (2008). Recuperado a partir de <https://www.uv.es/ciuv/cas/zcalculo/calculouv/>

Winter, B. (2011, august). Pseudoreplication in phonetic research. Hong-Kong.

Winter, B. (2013a). A very basic tutorial for performing linear mixed effects analyses.

Winter, B. (2013b). Linear models and linear mixed effects models in R with linguistic applications. Recuperado a partir de <http://arxiv.org/pdf/1308.5499.pdf>

Winter, B., & Grawunder, S. (2012). The Phonetic Profile of Korean Formality. *Journal of Phonetics*, 40(6), 808-815.